SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Comparisons of Filter, Wrapper and Embedded-Based Feature Selection Techniques for Consistency of Software Metrics Analysis

[1]Shamsuddeen M. Abubakar, [2]Zahraddeen Sufyanu, [3]Abdullahi M. Ibrahim, [4]Abubakar M. Miyim, [5]Vijay Arputharaj, and [6]Senthil Kumar A.
[1, 2, 3, 4] Department of Computer Science, Faculty of Computing Federal University Dutse, Jigawa State.
[5,6]Department of Computer Science, Skyline University Nigeria.
salsabil012@gmail.com, sufyanzzzzz@gmail.com, abdul_ami@yhoo.com, abubakar.miyim@fud.edu.ng, phdvij@gmail.com, senthil.kumar@sun.edu.ng

Abstract

Identifying and selecting the most consistent subset of metrics which improves the performance of software defect prediction model is paramount but challenging problem as it receives little attention in literature. The current research aimed at investigating the consistency of subsets of metrics that are produced by embedded feature selection techniques. Ten (10) feature selection techniques used from the families of filter and wrapper-based feature selection techniques commonly used in the defect prediction domain. Ten (10) publicly available defect datasets were studied which span both proprietary and open source domains. SVM-RFE-RF presented 42-93% consistent metrics across datasets. While the prior study on non-Embedded produced 56.5% consistent metrics at median. SVM-RFE-LF approach of Embedded Feature Selection Technique produced 54-80% consistent metrics across datasets and 42.5% at median. To state the purpose of tittle has been achieved Embedded based Feature Selection Techniques produced most efficient consistent subset selection across the entire datasets and amongst the feature selection techniques as compared with counterpart filter and wrapper-based feature selection techniques.

Keywords*: Defect Prediction, Software Metric, Filter-based, Wrapper-based and Embedded Feature Selection Techniques*

1. Introduction

Software Quality Assurance (SQA) teams played the most vital role in software development processes that ensured defect free software is delivered to end-users. In the developed software companies such as Amazon, Facebook, Mozilla, and Blackberry all have a separate SQA department that spearhead the quality and maintainability of software (Abubakar *et al.,* 2021a).

Buckley *et al.* (1984) stated that SQA and software engineers in any organization must concurrently perform the following task so as to deliver the most efficient software:

I. They design, implement, and execute tests to verify and validate that software systems satisfy their functional and nonfunctional requirements, as well as meet user expectations to ensure systems are of sufficient quality before their release to customers (Abubakar *et al.,* 2021b).

II. They also review designs, look closely at code Quality and risk, and refactoring code to make it more testable (Avram, 2011).

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Previous work raised concerns over the fact that SQA activities are expensive and time consuming (Tantitthamavorn *et al.,* 2018). According to Alberts *et al.* (1979) SQA activities requires almost 50% of the software development resources. Therefore, it is likely not possible to thoroughly test and review such a large software product given the limited SQA resources in terms of team size and time. For example, Facebook allocates about 3 months to test a new product.

Project managers find it more essential to outline appropriate software quality improvement plans to mitigate the risk of introducing defects in future releases. These may save industries billions of dollars to find and correct the defective software module after the release (Abubakar *et al.,* 2021b).

According to Li *et al.* (2017), feature selection techniques should only be applied on training samples to avoid producing optimistically biased performance estimates and interpretation. Feature subset selection can be divided into three models: filters, wrappers and embedded. All feature selection models have their own advantages and drawbacks.

(I) Filter-Based Techniques

Filter methods were the earliest approach to feature selection in machine learning (Hall, 1998; Maimon, 2010). Filters are algorithms which filter out insignificant features that have little chance to be useful in analysis of data. They are fast due to the fact they do not incorporate learning and rely on the intrinsic characteristics of the training data to select and discard features.

(II) Wrapper-Based Techniques

Wrapper methods usually select feature subset by a learning algorithm as part of the evaluation function. The learning algorithm is used as a black box function which guide the searching of candidate subset. The evaluation function for each candidate feature subset returns an estimate of the quality of the model that is induced by the learning algorithm (Langley, 1994).

(III) Embedded-Based Techniques

In contrast to filter and wrapper, the learning part and the feature selection part are all combined together in an embedded method (Quinlan, 1986; Quinlan, 1992). Decision tree learning can also be considered as an embedded method. The construction of the tree and selection of the features are interleaved. Feature selection techniques may produce different subsets of metrics for each training sample that is randomly generated from model validation techniques. Such inconsistency of subsets of metrics often leads to different interpretation among training samples (Abubakar *et al.,* 2021a). However, there exist many researches that have chosen only two of the three family of feature selection techniques, namely: filter-based feature selection techniques and wrapper-based feature. For example, Jiarpakdee *et al.* (2018b) stated that their results may not be generalized to other feature selection techniques and also cited the negative impact of wrapper based feature selection techniques as computationally time consuming and not cost effective. While, Filter-based feature selection techniques are fast because they do not incorporate learning algorithms and rely on intrinsic characteristics of the training data to select and discard candidate feature set.

It was further stated by Tantithamvorn *et al.* (2019), that previous researches made re-use of NASA datasets and Anova Type-I datasets in software defect models. As using such datasets generate noise and lead to misleading defect model construction and interpretation. The current study is set to exclusively investigate the consistency of subset of metrics that are produced by Embedded-based feature selection techniques family with the aim at making critical analysis between the families. Also, the research will avoid using NASA datasets and Anova Type-I datasets over the said issues. To highlight the importance of defect model interpretation as stated in Jiarpakdee *et al.* (2018b) it was believed that consistent software metrics selection provided a solution to the problems leading to misleading information on defect prediction interpretation. It was however believed that feature selection techniques may produce subsets of inconsistent and correlated metrics if quality software metrics were not used in the construction

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

of defect model. This has mandated us to conduct a thorough investigation and make analysis on the consistency of the subset of metrics produced that are generated by embedded feature selection techniques.

2. Related Works

Literatures related to the research topic were gathered which resulted in finding the research gap. Gao *et al.* (2015) compare between Feature Selection Techniques with Data Balancing techniques to identify the best for defect prediction model construction and interpretation. Feature Selection Technique was initially imported followed by Data Balancing. Feature Selection Techniques was found to have better prediction accuracy than Data Balancing. However, software metrics for consistency and correlation analysis was not covered in the research.

Work of Muthukrishnan *et al.* (2016) made comparisons between the three most popular Feature Selection regression algorithms, OLS Regression, Ridge Regression and the LASSO Regression. Real data and simulated environment using R packages was used in testing the performance of these procedures. Different algorithms were used in the research to compare the prediction accuracy of each algorithm in software defect prediction. But the algorithms were not tested in an Embedded Feature Selection Technique.

Xu *et al.* (2016) compared the impact of some Feature Selection Techniques on two versions of NASA datasets. The datasets comprise of noisy and clean and one open source AEEEM dataset. The research finding showed that Feature Selection Techniques had significant effects over all the datasets. The research looked into only Filter-based and Wrapper-based Feature Selection Techniques, while Embedded Feature Selection was not used over the said datasets.

Work of Nam *et al.* (2017) identified 50 instances of different datasets to be enough in building a defect prediction model. The research found that, when a software module used different software metric sets, it is possible to transfer these metrics into another software module in identifying defect prediction. The research never looked into testing the correlation between the metrics and consistency between the subset of metrics.

Work of Chatterjee *et al.* (2018) developed a model of defect proneness of the software modules where Mahalanobis distance metric was used in identifying defect prone software modules. During software development processes, project managers can identify where the defect prone software modules are. Performance analysis has shown better prediction accuracy of the model as compared to existing similar models. Process metrics, product metrics and organizational metrics were not covered in the research. NASA datasets was used in the conduct of training which reduces the prediction accuracy of the model.

Ndenga *et al.* (2018) compared Logistic Regression and Linear Regression algorithms to predict bug status and number of bugs corresponding to a file in a software module. The prediction performance of these model was measured against numerical and graphical prediction model. The research findings showed that change burst metrics contain the best numerical performance measures and have the highest defect detection probability. Only Filter-based and Wrapper-based Feature Selection Techniques were tested in the two algorithms.

Work of Thiruvathuka *et al.* (2018) developed a new algorithm called Metrics Dashboard, which was used for producing and observing metrics that are generated from open source software repositories on GitHub. The approach was used to allow users submit the URL of a hosted repository for batch analysis. User can interactively study various metrics over time numerically and visually.

The research also created a repository for obtaining datasets for training purposes. No consistency and correlation analysis on a subset of software metrics was discovered in the research. Dam *et al.* (2018) developed a new algorithm called Novel Prediction Model which automatically learned features to represent source code and use them for defect prediction. Evaluation on two datasets, one from open source projects which was contributed by Samsung and the other from the public PROMISE repository were performed. The effectiveness for both within-project and cross project predictions was determined. Consistency and correlation analysis were not taken into consideration.

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

The research of Das (2018) examined the advantages and disadvantages of both Filter-based and Wrapper-based Feature Selection for feature selection and proposed a hybrid algorithm that uses boosting and incorporates some of the features of wrapper methods into filter method for feature selection. The research did well in investigating the difference between filter and wrapper thereby coming up with hybrid algorithm and the result showed that wrapper is faster than filter feature selection techniques, but discarding the simple fact that embedded method is faster than both of the filter and wrapper methods.

The main objective of the research work of Huda *et al.* (2018) was proposing two novel hybrid software defect prediction models to identify the significant software metrics using a combination of Wrapper-based and Filter-based techniques.

The research findings include combining different Wrapper-based Techniques with Support Vector Machine and Artificial Neural Networks using relevance of Filter-based Technique to find the significant metrics. Hybrid heuristic algorithm can increase the consistency of metric selection. The research looked into Filter-based and Wrapper-based Techniques to test the new algorithm. Embedded-based Feature Selection was not tested using the proposed algorithm. The research also never looked into consistency and correlated of subset of metrics that can guide in improving quality of defect model construction and interpretation.

Abubakar and Sufyanu (2019) reported various algorithms used in defect prediction models. Software defect collection, defect model construction and model validation were all discussed. Meanwhile, literatures related to software defect are highlighted. When the defect models were compared, significance results were reported over previous feature selection techniques. The research suggested defect model using Embedded Feature Selection for consistency analysis.

Abubakar *et al.* (2020) reviewed the basic Feature Selection Techniques for Software Defect Prediction Model and their domain applications. Support Vector Machine with Recursive Feature Elimination for both Logistic Regression and Random Forest were introduced when evaluating the performance of filter, wrapper, and embedded feature selection techniques. The research proposed using Embedded Feature Selection techniques for correlation of a subset of software metrics analysis.

According to the work of Madeyski *et al.* (2021) several process metrics were compared with the product metrics in order to identify those that can improve the prediction accuracy of a defective software module. The research findings showed that product metrics that contains a distinct committers have better prediction accuracy than process metrics. The research was only limited to Filter-based and Wrapper-based Feature selection techniques as such the result may not be generalized to other organizational metrics. The current research will look into the improvement of quality software defect models by mitigating the correlated software metrics in the defect dataset using Embedded Feature Selection Techniques.

Abubakar *et al.* (2021a) compared the performance of filter-based and wrapper-based feature selection techniques with that of embedded-based feature selection technique. SVM-RFE-LR produced 18% to 55% consistent software metrics across the datasets, and SVMRFE-RF produced between 33% to 96%. Significant performance was recorded over Filter and Wrapper based techniques.

Research of Abubakar *et al.* (2021b) proposed the use of embedded feature selection techniques to mitigate and identify the correlated software metrics. Variable Clustering Function analysis was deployed using VarClus of the Hmisc in the R package of Rstudio. Support Vector Machine for Recursive Feature Elimination with Logistic Regression generated correlated metrics of 0.21 to 0.71, with 0.52 at median. Whereas, Support Vector Machine for Recursive Feature Elimination with Random Forest generated correlation within 0.29 to 0.74, with 0.39 at median. These have shown significant reduction of about 92% in the correlation analysis as against 86% previously reported.

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Table 1: Summary of Related Literatures

S/N	**Author**	**Method**	**Strength**	**Weakness**
1	Bhalaj *et al.,* 2022	Compares between classifiers of software defect models	Performance of feature selection methods when exposed to different datasets were tested	Filter, Wrapper and Embedded based methods were not compared
2	Tantithamthavn *et al.,* 2021	Logistics Regression and Random Forest algorithms were tested	Impact of class rebalancing technique and the interpretation of defect prediction models were investigated	Embedded based presents a better results as compared with other methods
3	Chatterjee *et al.,* 2021	Mahalanobis Distance Metrics Algorithm was developed	Increased the prediction accuracy	Comparison between the three techniques were not conducted
4	Das 2022	Filters and wrapper-based Feature Selection Techniques was compared	Difference between Filters and Wrapper based was investigated	Embedded based Feature Selection Techniques was not investigated
5	Abubakar & Sufyanu 2019	Recursive Feature Elimination with SVM was used	Embedded based feature selection techniques were deployed	comparisons between the three feature selection techniques were not conducted
6	Abubakar *et al.,* 2020	Recursive Feature Elimination with SVM was developed in testing the methods	Embedded-based Feature Selection Technique literatures was analyzed	Performance evaluation between the three feature selection techniques were not investigated

Having reviewed and analyzed various literatures related to the software defects prediction, it was found that in all the arguments presented by different researchers none has provided a better solution to consistent software metrics analysis which helps in selecting best subset of metrics used in software defect models construction. Therefore, the researcher found research gap in the various methods conducted by other researchers and research topic titled "Comparisons of Filter, Wrapper and Embedded-Based Feature Selection Techniques for Consistency of Software Metrics Analysis" believed will provide a solution to increasing the consistency and subset of metrics selection which help in producing better model performance.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

3. Proposed Approach

3.1 Data collection

In this study, the datasets were obtained from various proprietaries and open source domain after critical analysis and applying criteria in choosing them, their descriptions are shown in table 1.

Table 1: The ten (10) studied datasets.

Project	Dataset	Number of Module	Number of Metric	Defective Ratio %	EPV	AUC_{LR}	AUC_{RF}
Apache	Xalan2.6	885	20	46	21	0.79	0.85
Eclipse	Debug 3.4	1,065	17	25	15	0.72	0.81
	JDT	997	15	21	14	0.81	0.82
	Mylyn	1,862	15	13	16	0.78	0.74
	Platform 2	6,729	32	14	30	0.82	0.84
	Platform 2.1	7,888	32	11	27	0.77	0.78
	Platform 3	10,593	32	15	49	0.79	0.81
	SWT 3.4	1,485	17	44	38	0.87	0.97
Proprietary	Prop 1	18,471	20	15	137	0.75	0.79
	Prop 2	23,014	20	11	122	0.71	0.82

3.2 Experimental Set-up

The stages involved in achieving the aim and objectives include:

i. Go to the RStudio, packages are downloaded and installed. Devtools were installed using: install.package("devtools"), FSelector was installed using the command install.package("FSelector"), caret package was installed using: install.package("caret"), Hmisc was installed using: install.package("Hmisc") and Rnalytica was installed using the command devtools::install_github('software-analytics/Rnalytica').

ii. Load Libraries will load FSelector library using: Library("FSelector"), Hmisc was loaded using: Library("Hmisc"), caret was loaded using: Library("caret"), Rcor was loaded using Library("Rcor") and Rnalytica was loaded using: Library("Rnalytica").

3.2 Training the Datasets.

Training for Consistency across Datasets which was categorized into four and consistency across Feature Selection Techniques which is SVM-RFE-LR and SVM-RFE-RF.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

a. Training for Consistency across Datasets for the four categories:

Training the consistency across Dataset for the first category was employed using:
set.seed(1)indices.1 < - sample(nrow(dataset)) cfs.metric.subset.1 < - sort(cfs(as.formula (paste(dep))))
Training the consistency across Dataset for the second category was deployed using
set.seed(2)indices.2 < - sample(nrow(dataset)) cfs.metric.subset.2 < - sort(cfs(as.formula (paste(dep))))
Training the consistency across Dataset for the third category was generated using
set.seed(3)indices.3 < - sample(nrow(dataset)) cfs.metric.subset.3 < - sort(cfs(as.formula (paste(dep))))
Training the consistency across Dataset for the fourth category was employed using
set.seed(4)indices.4 < - sample(nrow(dataset)) cfs.metric.subset.4 < - sort(cfs(as.formula (paste(dep))))

b. Consistency across Feature Selection Techniques

Training consistency across SVM-RFE-LR was loaded using:

```
lrFuncs_AUC <- lrFuncs
lrFuncs_AUC$summary <- twoClassSummary
control <-
  rfeControl(
    functions = lrFuncs_AUC,
    method = "boot",
    number = 100,
    verbose = T
  )
set.seed(1)
results.lr.1 <-
 svm rfe (
    dataset [indices, indep],
    dataset [indices, 'depFactor'],
    sizes = c (1: length(indep)),
    rfeControl = control,
    metric = 'ROC'
  )
Svm.rfe.lr.metric. subset.1 <- sort(predictors(results.lr.1))
svm.rfFuncs_AUC <- rfFuncs
```

Training consistency across SVM-RFE-LR was loaded using:

```
rfFuncs_AUC$summary <- twoClassSummary
control <-
  rfeControl(
    functions = rfFuncs_AUC,
    method = "boot",
    number = 100,
    verbose = T
```

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

```
 )
set.seed(1)
results.rf.1 <-
 svmrfe (
   dataset [indices, indep],
   dataset [indices, 'depFactor'],
   sizes = c (1: length(indep)),
   rfeControl = control,
   metric = 'ROC'
 )
Svm.rfe. rf. metric. subset.1 <- sort (predictors (results.rf.1))
```

Consistency across Feature selection and across datasets will generate the results, PSPP was used in making the analysis. Microsoft Excel was used to compute the percentage consistency while Past325 was used in plotting boxplot. Figure 1 shows the processes involved in achieving the research objectives.

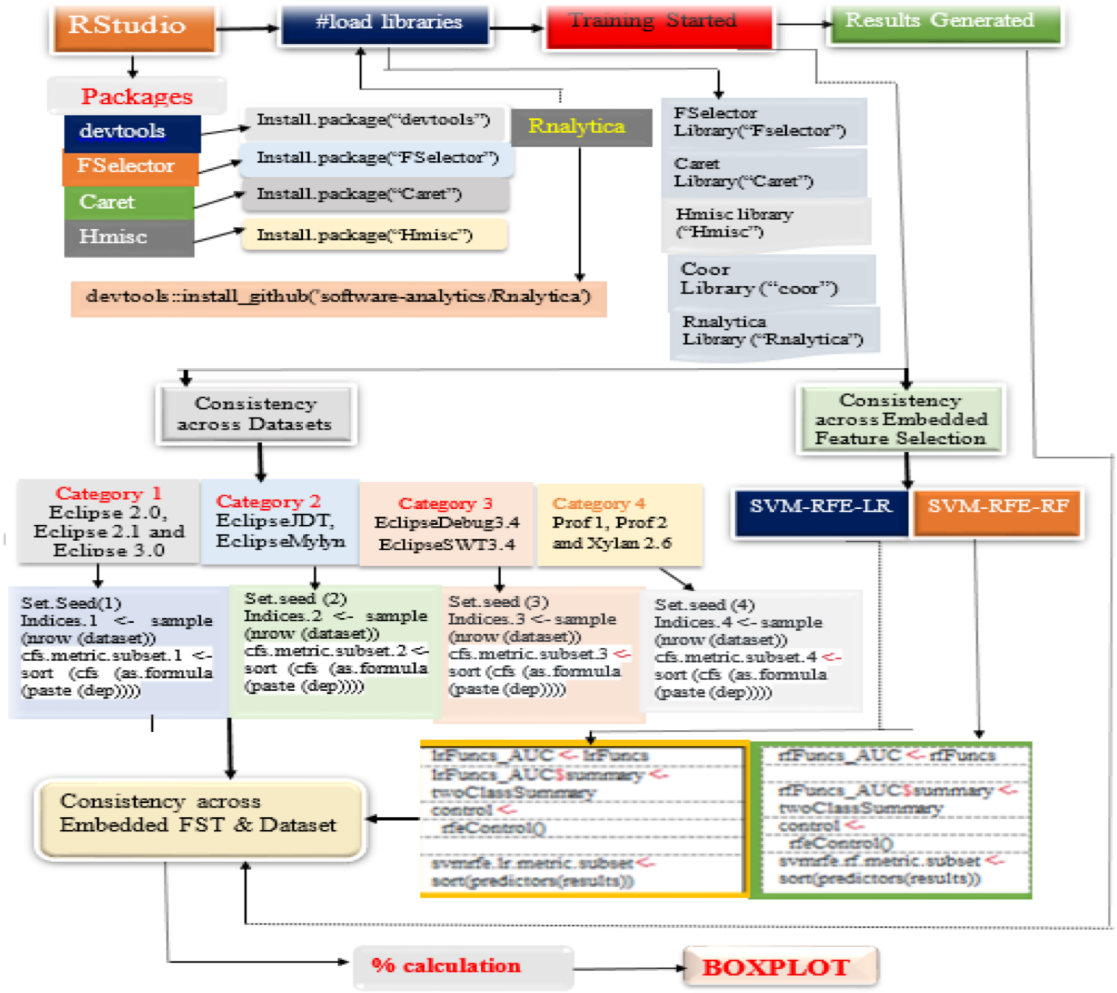

Fig. 1: General experimental set-up

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Results obtained for consistency across Feature Selection Techniques and consistency across Dataset are taken into PSPP for Data Analysis. Stages involved are described as follow:

a. The obtained results was in a format True and False, which are changed to 1 and 0 respectively.

b. Logical operations, AND Gate and OR Gates function from PSPP were used

i. For AND Gate

Mi1 and Mi2 and Mi3 and Mi3……..... Mij

Mj1 and Mj2 and Mj3 and Mj3……..... Mij

ii. For OR Gate

Mi1orMi2orMi3orMi3……. Mij

Mj1or Mj2orMj3orMj3……. Mij

Figure 2 will present the implementation of PSPP and various stages involved in computing the percentage consistency.

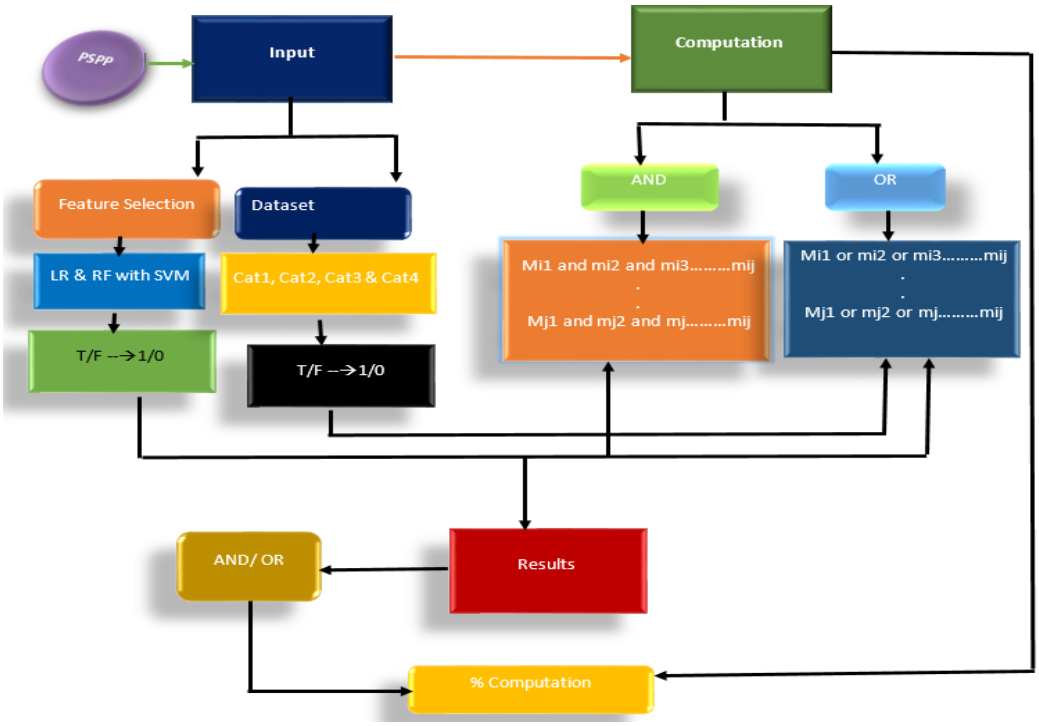

Fig.2: PSPP Implementation

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

The logical operands where used to compute the intersection and union for the results, consistency across feature selection Techniques and consistency across Datasets. Figure 3 describes how AND and OR gates where used to generates Intersections and Unions for percentage calculations.

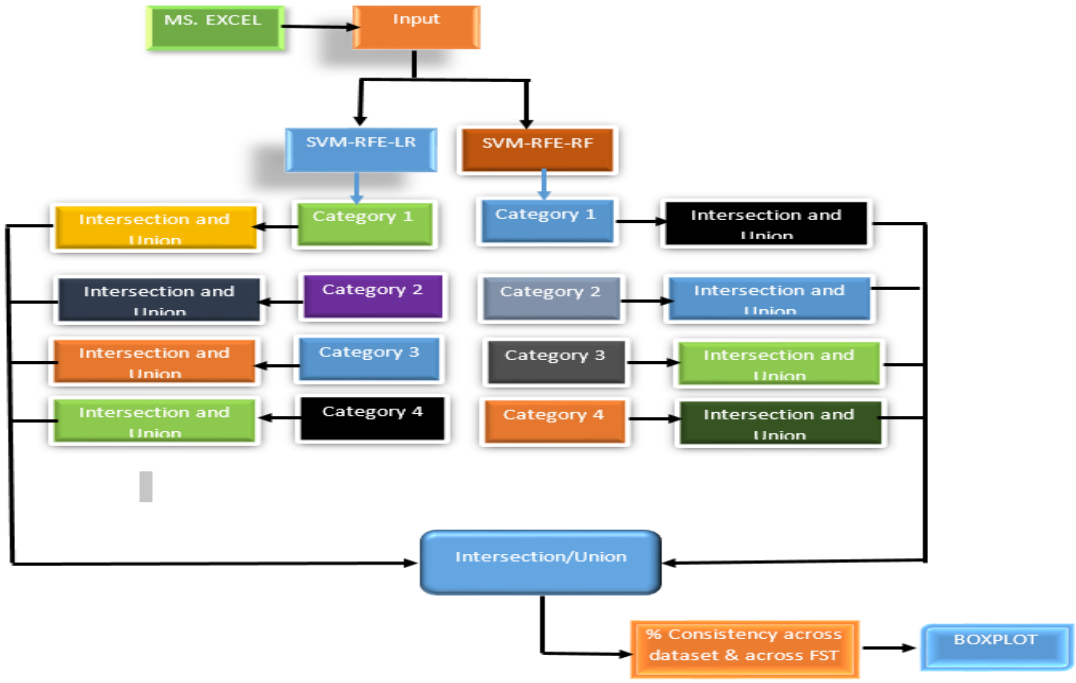

Fig.3: Microsoft Excel Implementation

Finally, the boxplots which is used in presenting the results were generated using the processes described in Figure 4.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

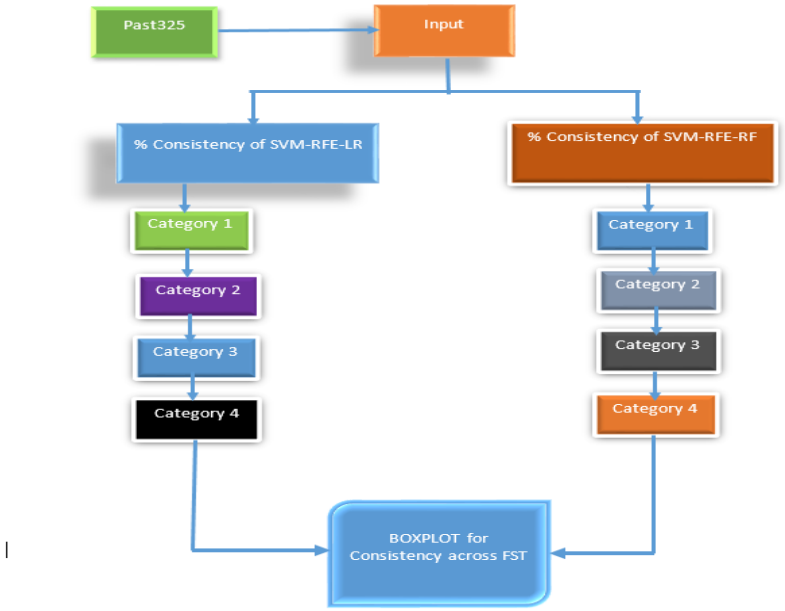

Fig.4: Past325 Implementation

4 Results and Discussions

Results for Support Vector Machine with Recursive Feature Elimination for both Logistic Regression (SVM-RFE-LR) and that for Support Vector Machine with Recursive Feature Elimination for Random Forest (SVM-RFE-RF) will be presented. Initially, consistency across feature selection techniques were tested on each of (SVM-RFE-LR) and (SVM-RFE-RF). Finally, consistency across the entire datasets were tested on (SVM-RFE-LR) and (SVM-RFE-RF) and their results were compared with that of previous studies.

1. Consistency across Feature Selection Techniques

a. SVM-RFE-LR

The results shown are the metrics selected by SVM-RFE-LR Embedded Feature Selection technique, and it showed either TRUE/FALSE format. Datasets with similar metrics were categorized the into category one (1) through four (4).

Category 1: These are the datasets which include Eclipse 2.0, Eclipse 2.1 and Eclipse 3.0. Table 2 shows the matrices selected by the SVM-RFE-LR approach of Embedded Feature Selection technique.

Category 2: These are the datasets which include EclipseJDT and EclipseMylyn. Table 3 below will present the matrices generated by SVM-RFE-LR approach of Embedded Feature Selection technique for the second category datasets.

Category 3: These are datasets which include EclipseDebug3.4 and EclipseSWT3.4. In table 4 below matrices produced by SVM-RFE-LR approach for the third category of datasets will be presented.

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Category 4: The datasets of the category consist of Prop 1, Prop 2 and Xylan 2.6. Table 5 will demonstrate the matrices produced by SVM-RFE-LR approach of Embedded Feature Selection technique for the fourth category of datasets.

b. SVM-RFE-RF

The results produced by Support Vector Machine with Recursive Feature Elimination for Random Forest will be illustrated. The subsequent tables will demonstrate the metrics generated by SVM-RFE-RF Embedded Feature Selection technique, the results are in the TRUE or FALSE format which will later be changed to 1 or 0 respectively.

Category 1: In this category of datasets we present eclipse 2.0, eclipse 2.1 and eclipse 3.0. From table 6 up to table 9 metrics selected by SVM-RFE-RF approach of Embedded Feature Selection technique for the first category of datasets are represented.

Category 2: In this category of datasets Eclipse JDT and Eclipse Mylyn are tabulated. Table below describe the metrics generated by SVM-RFE-RF for the second category of datasets.

Category 3: In this group datasets include EclipseDebug 3.4 and EclipseSWT 3.4. Table 8 shows the metrics selected by SVM-RFE-RF for the third category of the datasets.

Category 4: These are the datasets which include Prop1, Prop 2 and Xylan 2.6. Table 9 shows the matrices selected by SVM-RFE-RF approach of Embedded Feature Selection technique for the fourth category of datasets. All the respective tables are shown in appendix i.

2. Consistency across Datasets

To measure the consistency of subset of metrics across the datasets we calculated the datasets obtained in each category. In each category metrics with similar characteristics are compared as demonstrated in the previous section.

For better computation of consistency, the results were obtained in the form of True and False which were changed into logical form of 1 and 0, True and False respectively.

I. SVM-RFE-LR

The metrics produced by SVM-RFE-LR are presented based on their categories and are shown in the format 0/1.

Category 1: Here we present the matrices selected by SVM-RFE-LR which includes Eclipse 2.0, Eclipse 2.1 and Eclipse 3.0. The corresponding outcome for intersection of three datasets shows either 1 or 0. When all the metrics selected turn up to be 0, then their union will show 0, else, it will be 1. The intersection and union obtained from this represented as 0 and 1 are also recorded in table 10, meanwhile, the total number of metrics are also calculated in the table.

Category 2: The category consists of the datasets which includes EclipseJDT and EclipseMylyn. The table is shown at appendix ii which represents the matrices selected by SVM-RFE-LR for the Second category of datasets. The corresponding intersection and union are recorded with total metrics selected.

Category 3: These are the datasets which consist of EclipseDebug3.4 and EclipseSWT3.4. Table 12in appendix ii represents the matrices selected by SVM-RFE-LR for the third category.

SLUJST
Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Category 4: These are the datasets which include Prop1, Prop 2 and Xylan 2.6. Table 4.13 in appendix ii shows the represented matrices selected by SVM-RFE-LR for the fourth category.

The consistency for SVM-RFE-LR are computed from the results obtained from each of the category and their percentages were also measured.

Table 14 in appendix ii presents the general results of Support Vector Machine with Recursive Feature Elimination for Logistic Regression for all the four categories of the datasets.

Table 14: SVM-RFE-LR Consistency

Category	Intersection	Union	Percentage
Category 1	16	29	55.17241379
Category 2	7	13	53.84615385
Category 3	8	10	80
Category 14	15	20	75

From the table it can be deduced that SVM-RFE-LF approach of Embedded Feature Selection Technique produced 54-80% consistent metrics across datasets and 42.5% at median. Were the previous research on filter and wrapper-based feature selection techniques produced only 6% consistent metrics at median. These has shown a greater performance as compared to previous researches.

II. SVM-RFE-RF

The metrics selected by SVM-RFE-RF approach of Embedded Feature Selection technique are represented based on the category of datasets which are demonstrated in 0 or 1 formats.

Category 1: This shows the representation of metrics selected by SVM-RFE-RF for the first category of datasets. The metrics selected for all datasets in this category and metrics selected for at least one dataset (intersection and union respectively) are recorded. The total metrics selected for both union and intersection as shown in Table 15 in appendix ii.

Category 2: This shows the matrices selected by SVM-RFE-RF for the second category of datasets. The metrics selected for all datasets in this category and metrics selected for at least one dataset in the same category (intersection and union respectively) are recorded. The total metrics selected for both union and intersection as shown in Table 16 at appendix ii.

Category 3: In this category the matrices selected by SVM-RFE-RF for the third category of datasets are represented. The metrics selected for all datasets and metrics selected for at least one dataset (intersection and union respectively) are recorded with total metrics selected for both union and intersection as shown in Table 17 in appendix ii.

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Category 4: This category shows the matrices selected by SVM-RFE-RF for the fourth category of datasets are represented. The metrics selected for all datasets in this category and metrics selected for at least one dataset in the same (intersection and union, respectively) are recorded with total metrics selected for both union and intersection as shown in Table 18 in appendix ii.

Following explain the consistency of SVM-RFE-RF computed from the results demonstrated in the previous sections. Table 19 presents the general results generated for support Vector Machine with Recursive Feature Elimination for the four categories of datasets.

Table 19: SVM-RFE-RF Consistency

Category	Intersection	Union	Percentage
Category 1	28	30	93.333333
Category 2	8	15	53.333333
Category 3	9	13	69.230777
Category 14	8	19	42.105267

From the table above SVM-RFE-RF produced 42-93% consistent metrics across datasets. While the previous study on filter and wrapper based feature selection techniques produced 56.5% consistent metrics at median. In the figure 1 below boxplot showing the consistency across the datasets for embedded feature selection Techniques with Logistic Regression was presented. Table 20 will give full description and explanation of the plot. Also in Figure 5 the boxplot will be showcasing the consistency across the datasets for embedded feature selection Techniques with Random forest against the percentage consistency. Table 21 will give full description and details of the plot.

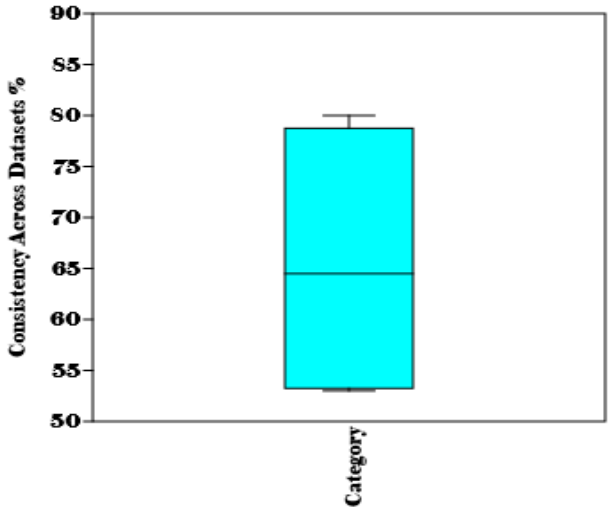

Fig.5: Consistency Across Datasets SVM-RFE-LR

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Table 21.: Description of Boxplot details

All	Lower conf.	Upper Conf.	
N	4	4	4
Min	53		
Max	80		
Sum	262214 310		
Std. error	7.00595	0.25	7.652614
Variance	196.3333	0.25	234.25
Stand. dev	14.0119	0.5	15.30523
Mean	65	553.5	77.5
Median	64.553	80	
25 percentiles	53.25	53	75
75 percentiles	78.75	54	80
Skewness	0.1046891	-2	2
Kurtosis	-5.349362	-6	4
Geom. mean	64.37322	53.49766	77.45967
Coeff. Var	21.39221	0.9302326	22.92918

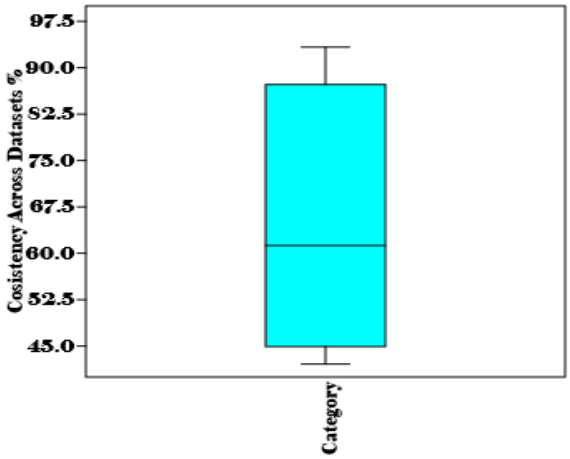

Fig.6: Consistency Across Datasets SVM-RFE-RF

Sule Lamido University Journal of Science & Technology Vol. 4 No. 1&2 [July, 2022], pp. 188-204
https://doi.org/10.56471/slujst.v4i.238
Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

It is practically proven that Embedded based feature selection techniques have the highest percentage of consistently selected metrics as compared with its counterpart Filter and Wrapper based feature selection techniques, and as such will improve the prediction accuracy of the software defect prediction model which subsequent research will look onto it. Figure 6 will show the comparative relationship between the embedded based feature selection techniques with its counterpart filter and wrapper based feature selection techniques.

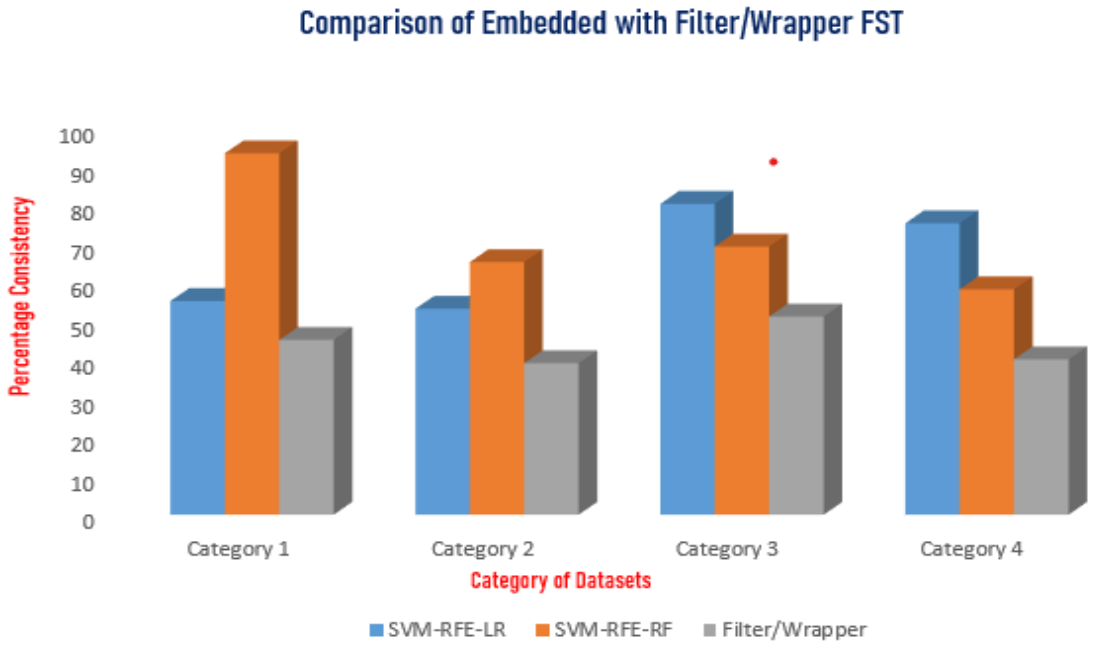

Fig.7: Comparison of Embedded with Filter/Wrapper

Conclusion and Recommendation

Embedded based feature selection can be efficiently used when selecting a subset of metrics compared to its counterpart filter and wrapper-based feature selection techniques. Consistency of subset of metrics selection impact the performance of a defect prediction model. Change in the datasets used for similar work. Changing the types of datasets will provide justification of embedded feature selection technique as best feature selection for defect prediction model, as well allow for more datasets exploration and in-depth assessment and rating scales to assess filter, wrapper and embedded feature selection technique for defect prediction models.

References

Abubakar, S, M., Sufyanu Z. and Miyim, A., M. (2019). Proposed Defect Modelling for Mitigating Correlated Software Metrics. Dutse Journal of Pure and Applied Sciences. Vol. 5 (2b): 76-86.

Abubakar, S., M. and Sufyanu Z. (2019). A Survey of Feature Selection Techniques for Software Defect Prediction Model. Federal University Dutsinma Journal of Science. Vol. 3, No 4: 258-265.

Online ISSN: 2736-0903
Print ISSN: 2736-089X
www.slujst.com.ng

Abubakar, S., M., Sufyanu Z., and Garko A., B. (2021a). Performance Evaluation of Embedded Feature Selection Techniques Over Filter and Wrapper-Based Feature Selection Techniques. Sule Lamido Journal of Science and Technology. Vol.2 (2): 33-45

Abubakar, S., M., Sufyanu Z., and Garko A., B. (2021b). Impact of Correlated Software Metrics on Embedded Feature Selection Techniques. International Journal of Information Processing and Communication (IJIPC) Vol. 11 (2) 118-134

Avram, A. (2021). Ensuring product quality at google. https://www.infoq.com/news/2011/03/Ensuring-Product-Quality-Google.

Das, S. (2022). Filters, Wrappers and Boosting-based Hybrid for Feature Selection. Division of Engineering and Applied Science Harvard University, Cambridge, MA 02B8, USA.

Gao, K., Khoshgoftaar, T. M., and Seliya, N. (2022) Predicting high-risk Program modules by selecting the right software measurements. Software Quality Journal, 20(1):3–42.

Jiarpakdee, J., Chakkrit, T., and Ahmed E., H. (2018). The Impact of Correlated Metrics on Defect Models. IEEE Transactions on Software Engineering.

Jiarpakdee, J., Chakkrit, T., and Christoph, T. (2018). Autospearman: Automatically Mitigating Correlated Software Metrics for Interpreting Defect Models. *IEEE International Conference on Software Maintenance and Evolution (ICSME)*.

Jiarpakdee, J., S., Y., Thongtanunam, P., and Tantithamthavorn, C. (2019). Mining Software Defects: Should We Consider Affected Releases? In Proceedings of the International Conference on Software Engineering (ICSE).

Kirbas S., B. Caglayan, Hall, T., Counsell, S., Bowes, D., Sen, A. and Bener, A., (2017). the Relationship between Evolutionary Coupling and Defects in Large Industrial Software. Journal of Software: Evolution and Process, 29(4):20-45

Liu C, Jiang, D., and Yang, W. (2014). Global geometric similarity scheme for Feature Selection in fault diagnosis. Expert Systems with Applications, 41(8): 3585–3595. International Master's Thesis, Department of Technology at Orebro University, 2012.

Nam, J., Weifu, Singhum, K., Tim, M. and Lin, T. (2017). Heterogeneous Defect Prediction. IEEE Transaction on Software Engineering.

Ndega, K., Malaga, I., G., and Mehat, F. (2018). Performance and cost-effectiveness of change burst metrics in predicting software fault. Springer –verlaglando ltd, post of springer Nature.

Okutan, A. (2020). Use of Source Code Similarity Metrics in Software Defect Prediction. arXiv: 1808.10033v1 [cs.SE].

Osman, H., Mohammad, G., and Oscar, N. (2018). the Impact of Feature Selection on Predicting the Number of Bugs. arXiv: 1807.04486v1 [cs.SE].

Shepperd, M., Song, Q., Sun, Z., and x Mair, Z. (2013). Data Quality: Some CommentsOn the NASA Software Defect Datasets. Transactions on Software Engineering (TSE), 39(9):1208–1215.

Xu, Z., Jin, L., Zijiang, Y., and Xiangyang, J. (2016). The Impact of Feature Selection on Defect Prediction Performance: An Empirical Comparison