ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SOFTWARE FAULT PREDICTION SYSTEM USING MACHINE LEARNING TECHNIQUE

Rupam Kumari[*1], Vijaybhan[2] & Ramesh Kumar Mohapatra[3]
[*1,2&3]Department of Computer Science and Engineering, National Institute of Technology, Rourkela India – 769008

ABSTRACT

Various Classification techniques have been ex-plored by the different researchers previously for the prediction of a software fault. It is noticed that the result of several technique changes for software to software and no one technique has always given a good result in all datasets. Moreover, Ensemble methods take the benefits of different individual prediction techniques and produce a better performance as compared to a single technique. Most of the works are available to classify the software system whether it is having fault or non-fault but very few methods are present that can caught the faults using ensemble techniques. The main objective of presented system to count faults present in the software. We have implemented the most popular and widely used machine learning algorithms Linear Regression, Decision Tree, Random Forest, Ada Boost, Extra Tree Regression, k-Nearest Neighbour, Gradient-Boosting, Multi-Layer Perceptron, Bagging Regression, Bayesian Ridge Regression, Stochastic Gra-dient Descent, Support Vector Machine. The twelve Machine Learning techniques are implemented to find the better base learners for heterogeneous ensemble learning. We have selected the best 4 base learner bagging Regression, k-NN, Support Vector Machine and Random Forest for Stacking Regression to improve the performance of the model. 15 versions of data sets are used in this work. The presented work gives a uniform result on all the dataset. Several performance methods which measure the AAE, ARE and accuracy has been done to evaluate the presented heterogeneous technique. We have observed that in all dataset Stacking Regression gives a better result as compared to the selected 4 top machine learning technique. The proposed heterogeneous ensemble techniques provides an enhanced result compared to single regression technique. The principal consequence of our research is to provide greater use of limited testing resources is to fix the defects right on time and immediate recognition of greater part faults present in the software modules.

Keywords: *Number of faults, AAE, ARE, SVM, kNN, BR, SR*

I. INTRODUCTION

As per ANSI Software Reliability characterized as likeli-hood of failure free software execution for a predetermined time frame in a predefined domain. Software fault prediction is a vital assignment to enhance the nature of software. Future software issues rely upon past fault history data sets. As the demand for more speed execution and beneficial software system is growing frequently, their many-sided quality is likewise expanding constantly. Then again, guaranteeing the high calibre of software is an exorbitant assignment and requires an abundant measure of assets. Software testing is a fundamental however exorbitant movement to improved software development life cycle. Sufficient assets measures which provides to test the product framework completely. The developing interest for elite and productive software brings about the more mind-boggling frameworks and in this manner expands the likelihood of the number of faults. Then, it is for all intents and purposes impractical to test the product framework totally and completely with the restricted testing assets. The prediction of defective software modules in early stage may be exceptionally compensating streamlining in the endeavours is connected in the next phases of software improvement.

Prior various grouping methods have been utilized to per-form defect expectation for the given software frameworks. It incorporates different machine-learning procedures [9], for example, discriminant analysis [4], logistic regression, factor analysis, fuzzy classification, classification trees, a Bayesian network, artificial neural networks and support vector ma-chines etc.

However, the outcomes demonstrated that distinctive sys-tems have delivered diverse forecast execution and none of them has dependably given the best expectation comes about crosswise over various datasets collections. Also, the execution of strategies changes from datasets to datasets. Then again, numerous hypothetical and exact confirmations upheld which utilize the ensembles technique which can predict maximum defect outcomes. Group strategy takes the benefit of individual taking an interesting procedures for the given dataset and means to think of maximum fault outcomes. The issue in the fault prediction model is to find the global minima in the info work. Outfit strategy defeats this issue by joining the nearby minima of numerous blame forecast methods to give a better general estimate of the given info work. Gathering strategy guarantees to decrease the inadequacy of individual systems for enhanced blame forecast. Because of these advantages of utilizing outfit technique, numerous takes a shot at the use of group-based methodologies for paired class which find the number of faults in the given software. The upside of anticipating the count of flaws over foreseeing which will give the number of issues present in the software module and which will help the software tester to test the defective module first during testing phase.

In this paper, base learners is used to determine for the ensemble method, an experiment using twelve various fault prediction techniques and selected high performance 4 base learners for the ensemble method. In addition, we investi-gate one technique i.e Stacking Regression for heterogeneous technique. The base learners which has been used in this work are Bagging Regression, k-Neighbors Regression, Linear support vector regression and Random Forest Regression for the stacking regression. Our experiment is conducted on the faulty datasets.

The following work is organized as described below. The re-lated work and our proposed Software Fault Prediction Model are presented in Section II and Section III respectively. The proposed and feature extraction technique are also discussed in detail in the Section IV, and finally future work and conclusions are drawn in Section V.

II. RELATED WORK

Previous research gives the method for prediction the fault which is present in software module. Ruchika Malhotra [3] conducted a comparative study of 18 ML techniques using object-oriented metrics MLP, NB, AB, RBF, ADT, LMT, Bagging, LB and RF techniques (AUC greater than 0.7 in most of the data sets) for defect prediction. The results of the experiment showed that MLP technique gives better performance followed by NB and LR techniques but SVM and VP provides worst performance for prediction of defects. 18 ML techniques have been validated using 10-cross-validation and inter release models. Data pre-processing is done by using the Correlation-based Feature Selection to remove the irrelevant features. It is summarized from the results that inter release models perform better than 10 fold cross-validation. The statistical test and post-hoc analysis using Friedman and Nemenyi test to generalized the obtained result from the experiment analysis. Santosh Singh Rathore, Sandeep Kumar [9] They proposed a heterogeneous method which can count the faults using linear and nonlinear combination rule. To divide the data sets Remove fold filter is used and for resampling of data set they used SMOTE algorithm. The Gradient boosting regression and Linear regression are used to merge the base learner output. The result obtained from the experiment says the proposed model gives better accu-racy than previous studies. After the experiment, they found that the ensemble based system gives better performance as compared to Single prediction technique. Rathore, Santosh S and Kumar, Sandeep [7]: This comparative study presented an experimental evaluation of six different techniques to predict the software fault like negative binomial regression, linear regression, genetic programming, decision tree regression, multilayer perceptron and zero-inflated Poisson regression.

For fault prediction technique they performed Dunns multiple comparison tests and KruskalWallis test for comparing the re-sults. During the analysis of results, they observed that Multi-layer perceptron, decision tree regression, linear regression and genetic programming gives higher performance in the given data sets that have been used in this study. After analyzing the results of Dunns multiple comparison tests and KruskalWallis test they found that zero-inflated Poisson regression (ZIP) and negative binomial regression (NBR) techniques provided the worst performance than other software fault techniques.

Rathore, Santosh Singh and Kumar, Sandeep [8] in this experimental study they presented Decision tree regression (DTR) to find the number of software fault by using two different way inter-release prediction and intra-release (10-

ISSN 2348 – 8034
Impact Factor- 5.070

fold cross-validation) predictions for the software project. In comparison analysis, they concluded that in intra-release fault prediction methods performs better than inter release fault prediction method on all datasets. The predicted faults produced by the DTR is very much similar to the actual faults. So after analyzing results, they observed that DTR furnished better performance as compared to existing tech-niques. Rathore, Santosh Singh and Kumar, Sandeep [10] presented a technique for finding how many faults present in the software project releases. After analyzing the result they observed that presented system gives better accuracy than the single fault prediction techniques. The measure of completeness analysis and in level l Prediction proved that proposed model performance is effective and the results are uniform throughout all data sets. After observing the results of different combination rule on a different way of prediction they found that performance results are effective in case of linear combination rule of ensemble method as compare to nonlinear combination rule of ensemble method.

III. PROPOSED WORK

In this work, we have proposed a Software Fault Prediction System as shown in Fig.1 and Fig.2 in which we are using heterogeneous ensemble technique i.e stacking regression by using various machine learning techniques. In the presented work, first of all, we collected the datasets from the NASA software then we divided our datasets into training and testing where training dataset consists 2/3 of dataset and testing data remaining (1/3) of the dataset. After that, we normalized the dataset using Min-Max Normalization technique, for dimen-sion reduction Principle Component Analysis (PCA) is used to reduce the feature set for post-processing of the dataset. As shown in Algorithm 1 and in Algorithm 2 we describe the all the phases of our research work. It contains the different steps as shown in proposed algorithm. We have applied 12 different machine learning techniques on various 15 versions of PROMISE repository NASA software datasets then we checked the performance measure of 12 machine learning techniques using performance measure parameters.

After that, top 4 Machine Learning Techniques Bagging Regression, k-Neighbors Regression, Linear Support Vector Regression and Random Forest Regression which gives the

Algorithm 1: Proposed Phases

```
Input: D=Camel-1.6, Ant-1.7, Camel-1.4, Camel-1.2,
       Prop-3, Prop-2, Prop-1, Prop-5, Prop-4, Prop-6,
       Xalan-2.6, Xalan-2.5, Xalan-2.4, Xerces-1.4,
       Xerces-1.3
Output: AAE, ARE, Accuracy
Apply 12 machine learning methods M =Linear
  Regression, Decision Tree, Random Forest, Ada Boost,
  Extra Tree Regression, k-Nearest Neighbour, Gradient
  Boosting, Multi Layer Perceptron, Bagging
  Regression, Bayesian Ridge Regression, Stochastic
  Gradient Descent and Support Vector Machine
Normalize the Data using Min-Max Normalization
Split Datasets D into Dtrain and Dtest dataset
Apply PCA to select the features
forall each Dtrain, Dtest, M do do
  train the model M by Dtrain
  test the model M by Dtest
  Calculate AAE, ARE and Accuracy on each dataset
    Dtest
```

Algorithm 2: Proposed Phases Cont...

Input: D= Camel-1.6, Ant-1.7, Camel-1.4, Camel-1.2, Prop-3, Prop-2, Prop-1, Prop-5, Prop-4, Prop-6, Xalan-2.6, Xalan-2.5, Xalan-2.4, Xerces-1.4, Xerces-1.3

Output: AAE, ARE, Accuracy

1 Use 4 machine learning methods for base learner among 12 machine learning method which has lowest AAE and ARE M =Bagging Regression, Random Forest, kNN, SVM and Stacking Regression

2 Normalize the Data using Min-Max Normalization

3 Split Datasets D into Dtrain and Dtest data set 4

Apply PCA to select the features

5 meta-regression=min AAE of M

6 forall *each Dtrain, Dtest, M do* do

7 train the model M by Dtrain

8 test the model M by Dtest

9 Stacking Regression is used to combine the predictions of all M via a meta-regression Calculate AAE, ARE and Accuracy on each dataset Dtest

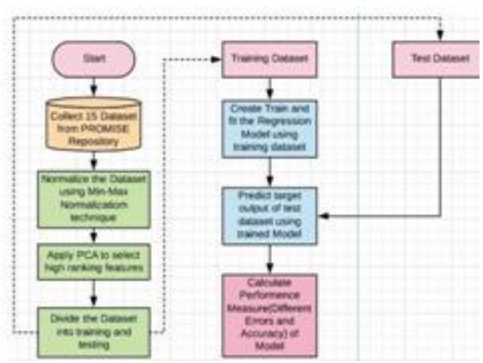

Fig. 1. Flow Chart of Software Fault Prediction System

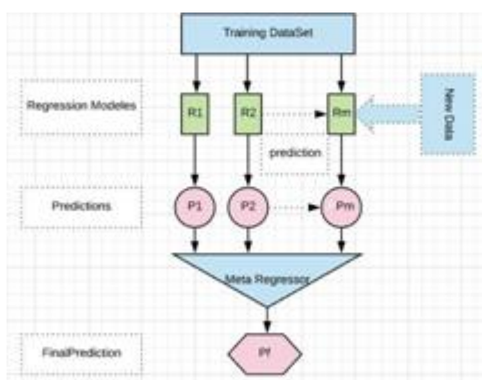

Fig. 2. Proposed working model of Software Fault Prediction System

better performance as compared to other techniques. To im-prove the performance we applied Stacking Regression tech-nique in which we chosen top 4 regression techniques as a base learner then we have trained the chosen techniques using training dataset and given the testing dataset to trained 4 top techniques to predict the faults then to combined the prediction results of these individual techniques meta-regression (mostly Linear Support Vector Regression) has been used which having lowest error and finally it gives the final prediction result.

IV. EXPERIMENTAL RESULTS

In this work we are collecting the dataset which are openly available on PROMISE data repository [1]. Our programming archive contains different types of software fault data which is taken from openly available software modules. As of now, it is having software fault data relating to three kinds of software attributes, CK Object-Oriented, Halstead or Maccabees met-rics, other static code metrics. We have utilized 15 versions of the 5 software to construct and assess the models which predicts the number of faults. In this paper we have first normalized the dataset which is having range between zero to one by using Min-Max normalization technique and then principle component analysis has been applied to select the important features and to reduce the high dimension of the used dataset. We observed from the shown Fig. 3 , Fig. 4 and Fig. 5 Stacking regression gives better results that are ARE=0.007 , AAE=0.006 and Accuracy=99.39% on PROP2 dataset w.r.t top four machine learning techniques and various used dataset to count the number of faults.

TABLE I TABLE COMPARISION OF RESULTS

Author name	ML Techniques	Preprocessing technique	Accuracy
Malhotra, Ruchika [3]	MLP, NB, AB, RBF, ADT, LMT, Bagging, LB, RF,	CFS	0.35 <= *AU C* <= 0.86
Rathore, Santosh Singh and Kumar, Sandeep [10]	Heterogeneous ensemble method and base learners-LR,GBR	Remove Fold Fil- ter and SMOTER algo	0.028 <= *AAE* <= 0.65 0.015 <= *AAE* <= 0.5
Rathore, Santosh S and Kumar, Sandeep [6]	Genetic programming technique	None .	0.08 <= *ARE* <= 0.48 25% <= *Recall* <= 65%
Wang, Tiejian and Zhang, Zhiwu and Jing, Xiaoyuan and Zhang,Liqiang [11]	Multiple kernel ensemble learning(MKEL)	Weight update for class imbalance	Pd=0.68,pf=0.26,f-measure=0.48
Rathore, Santosh S and Kumar, Sandeep [9]	GP, MLP,LR, DTR, ZIPR, NBR	10-fold cross validation Strati - fiedRemove Fold	0.107 <= *AAE* <= 2.57 0.06 <= *ARE* <= 1.15
Rathore, Santosh Singh and Kumar, Sandeep [8]	DTR	10-fold cross val- idation	0.156 <= *AAE* <= 0.85 0.6 <= *ARE* <= 0.9 27% <= *pred*(*l*) <= 84%
Rathore, Santosh Singh and Kumar,	Ensemble methods	RemoveFold and SMOT	0.4 <= *AAE* <= 0.7 0.3 <= *ARE* <= 0.55

ISSN 2348 – 8034
Impact Factor- 5.070

Sandeep [7]		algorit hm	E
Chen, Mingming and Ma, Yutao [2]	DTR, LR, GBR, SVR, NNR, GBR, Bayesian Ridge Regression	None	0.001 <= *P recision* <= 0.04 0.084 <= *Rmse* <= 0.970
Ayse Tosun Msrl, Basar Bener,Burak Turhan [5]	NB, ANN, Voting feature intervals	PCA	Accuracy=80% Pf=33%
Liguo Yu [12]	Negative binomial regression []	None	10% <= *recall* <= 44% 75% <= *accuracy* <= 90%
Proposed Work	LR, DT , RF, AB, ETR, KNN, GB, MLP, Bagging Rr, BRR, SGD,SVM,StackingRegressor	Min-Max Normalizatio n, PCA	93.6% <= *Accuracy* <= 99.4% 0.0067 <= *ARE* <= 0.073 0.006 <= *AAE* <= 0.0635 0.025 <= *RM SE* <= 0.126 0.0006 <= *M SE* <= 0.016

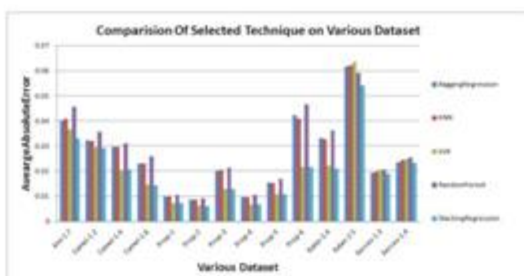

Fig. 3. Average Absolute Error of different classifier

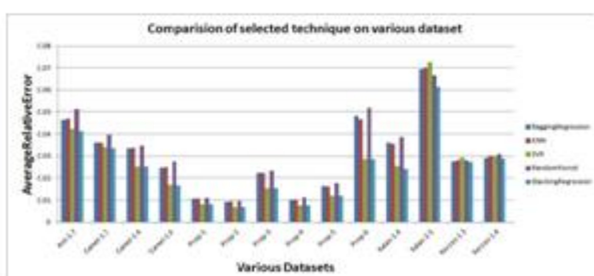

Fig. 4. Average Relative Error of different classifier

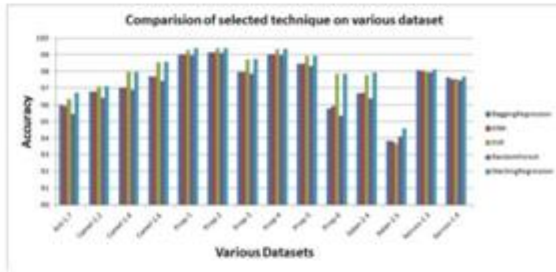

Fig. 5. Accuracy of different classifier

To evaluate the performance of proposed work Performance measure parameters such as AAE, ARE and accuracy has been considered and its equation are listed below:

$$AverageAbsoluteError(AAE) = \frac{1}{N}\sum_{j=1}^{N} |predicted_j - actual_j| \quad (1)$$

$$AverageRelativeError(ARE) = \frac{1}{N}\sum_{j=1}^{N} \left| \frac{predicted_j - actual_j}{actual_j + 1} \right| \quad (2)$$

$$Accuracy(\%) = 100 - ARE * 100 \quad (3)$$

where N is the number of data points, predictedj is the predicted number of faults in software module actualj is the actual number of faults in software module. In some cases, the value of $actual_j$ may be zero. So we have added 1 in the denominator to make the formula correct and generalized.

After observing the results of the experimental study we have investigated that Stacking Regression gives higher ac-curacy, less average absolute error and average relative error and performs well in all used data set. On an average, we got the value of ARE=0.02813, AAE=0.0244 and Accu-racy=97.5597 on entire dataset and Bagging Regression, k Nearest Neighbour, Support Vector Machine, Random Forest, Stacking Regression techniques. In comparison as shown in Table 1, our method gives the better result compare to others.

V. CONCLUSION AND FUTURE WORK

The main aim of our work is presented a heterogeneous technique for counting the faults present in the given software modules. First of all, we implemented different popular ma-chine learning algorithms for ensemble technique to choose the better prediction base learner for heterogeneous ensemble method. After analyzing the result we found that Bagging,k-NN, SVM and Random Forest gives less error and higher accuracy so we took these 4 learners as a base learner for Stacking Regression technique. During analysis of single prediction technique, we found that SVM presented better as compared to remaining three-technique then we used the SVM as combination method to merge the results of base learner technique in stacking regression ensemble technique. We performed the experimental study on 15 releases of dif-ferent NASA software modules that are open source software available at PROMISE Repository. Our experiment gives the better result for the predicting the faults present in the software modules. Different performance measure such as ARE, AAE, accuracy and Bar plot analysis of error and accuracy is also designed for better understanding of prediction results and come up with decision that proposed technique are more accurate and reliable. This proposed technique will reduce the effort and cost of software development life cycle. After performing the experiment on all dataset we compared and found the proposed result is better as compared to previous work. In Future Deep Learning and Neuro-fuzzy techniques can be used to improve the existing work. We can also analyze the presented system on large data set and on different domain software to generalize the result.

ISSN 2348 – 8034
Impact Factor- 5.070

REFERENCES

1. *G. Boetticher, "The promise repository of empirical software engineer-ing data," http://promisedata.org/repository, 2007.*
2. *M. Chen and Y. Ma, "An empirical study on predicting defect numbers." in International Conference on Software Engineering and Knowledge Engineering(SEKE), 2015, pp. 397–402.*
3. *R. Malhotra, "An empirical framework for defect prediction using machine learning techniques with android software," Applied Soft Com-puting, vol. 49, pp. 1034–1050, 2016.*
4. *G. McLachlan, Discriminant analysis and statistical pattern recognition. John Wiley & Sons, 2004, vol. 544.*
5. *A. T. Mısırlı, A. B. Bener, and B. Turhan, "An industrial case study of classifier ensembles for locating software defects," Software Quality Journal, vol. 19, no. 3, pp. 515–536, 2011.*
6. *S. S. Rathore and S. Kumar, "Predicting number of faults in soft-ware system using genetic programming," Procedia Computer Science, vol. 62, pp. 303–311, 2015.*
7. *R. S. S and K. Sandeep, "An empirical study of some software fault prediction techniques for the number of faults prediction," Soft Computing, vol. 21, no. 24, pp. 7417–7434, 2017.*
8. *R. S. Singh and S. Kumar, "A decision tree regression based approach for the number of software faults prediction," ACM SIGSOFT Software Engineering Notes, vol. 41, no. 1, pp. 1–6, 2016.*
9. *R. S. Singh and KumarSandeep, "Towards an ensemble based system for predicting the number of software faults," Expert Systems with Applications, vol. 82, pp. 357–382, 2017.*
10. *R. S. Singh and K. Sandeep, "Linear and non-linear heterogeneous ensemble methods to predict the number of faults in software systems," Knowledge-Based Systems, vol. 119, pp. 232–256, 2017.*
11. *T. Wang, Z. Zhang, X. Jing, and L. Zhang, "Multiple kernel ensemble learning for software defect prediction," Automated Software Engineer-ing, vol. 23, no. 4, pp. 569–590, 2016.*
12. *L. Yu, "Using negative binomial regression analysis to predict software faults: a study of apache ant," 2012.*