Comparative Analysis Of Heterogeneous Ensemble Learning For Software Fault Prediction

Jyoti Goyal, Bal Kishan

ABSTRACT:Software quality is the main aspect of every software product. Fault is the primary reason for decreasing the quality of software. Therefore we need fault prediction techniques that predict faults at early stage of software development. Ensemble learning has proved its significance in various scientific fields. So, in this paper, we propose heterogeneous ensembling technique because same kind of technique is not able to deal with all kinds of problems. Also we discuss various challenges that we still face during the development of the efficient fault prediction model using Ensembling.

Index Terms: Faults, heterogeneous ensembling, Machine Learning Techniques, software fault prediction

1. INTRODUCTION

The software industry is growing day-by-day because of its use in every field of life. But with the increased use of software, its complexity is also growing day-by-day. It is very difficult to test each module exhaustively because of the limited testing resources [1]. Therefore we need to prioritize the modules that are more fault prone so the testing team can use their scarce resources efficiently. This not only save the resources but also improves the efficiency of the system. A software fault prediction model solves this problem. In SFP, we use some specific features of the data to identify whether the module is faulty or not. These features are called metrics. These features are used to find the faults of the current project [2, 3]. Ample amount of work has been done in this field but no one proves itself to be better in all domains. Literature proves that no single technique is able to deal with all kinds of problems because of its sensitivity to specific type of problem [4, 5]. Also ensembling with same kind of techniques does not give promising result because these classifiers behave similarly in some portion of dataset [6]. So, in this paper we propose heterogeneous ensembling technique that is able to predict different kinds of faults in different data. Heterogeneity can be achieved in various ways, whether by using different data samples, by using different parameters of the algorithm, or by using different algorithm. In this paper we use multiple models of different ML algorithms to reduce generalization error. Here, we use stacking that works on two phases.in the first phase, different base classifiers are trained based on training data and the output of this phase acts as an input in the second phase. It means if a particular classifier consistently misclassifies defect data coming from a particular region, then the supervisor model will learn this behaviour and then correct the mistake. Also, stacking has better generalization capability [7]. One main problem with multiple models is over fitting but bagging takes care of it [8]. Now the question is what model to select for stacking. It is also a matter of art than science. We choose random forest, it has high predictive power. One advantage of stacked generalization, or “stacking” [7, 9], is simply alignment of classifier outputs with the definitions and parameters implicit in the training data collected for the application. In general, ensemble predictions improve when new inputs are added that are both informative and uncorrelated with existing ensemble components [10].

2. RELATED WORK

Prediction of faults using homogeneous ensembling techniques is done by various researchers. But prediction using heterogeneous ensembling is not highly researched. So Here we present the related work done in fault prediction by both of these ensembling techniques. Abdullah et al [11] compared various homogeneous ensembling techniques and machine learning techniques to know which perform better in different datasets and the results reveal that RF, bagging DS and Ada Boost performs better than other techniques. Shanthini et al. [12] compares the ensemble methods i.e bagging, Boosting, Stacking and Voting at various metrics levels. It is observed that bagging performs better for package and method level metrics and voting performs better for class level metrics. Chubato et al [13] performs an empirical and comparative study to deal with various data problems i.e. class imbalance, noise, and data skewness. Chubato proposed a three stage ensemble learning approach that resolve these data problems and improve the prediction performance of the model. Abdullateef et al. [14] proposed a novel evaluation method called ANP to evaluate the performance of the model. Accuracy is not the only parameter to judge the significance of any technique. Multi criteria decision making should be used. Rana et al. [15] presented a study called early prediction of faults that focuses on the early phases to predict the faults that really helps the testing team to utilize the testing resources. For this requirement and design metrics is used to predicts faults. Shamsul et al. [16] proposed a novel hybrid approach that deals with the class imbalance problem. This approach is a combination of random oversampling, majority weighted minority oversampling and fuzzy based feature instance recovery. This gives more accurate results than other standard classification techniques. Lov et al. [17] performs an empirical evaluation where ensemble models are used to perform prediction with reduced set of metrics. The results prove that MVE performs better with selected metrics and takes less cost. Xinli Yang et al. [5] performed a study that is very rarely discussed in literature. He proposed a novel approach i.e. TLEL a two stage approach for prediction of faults. In the inner layer, decision tree and bagging is used to build the random forest model and at the outer layer stacking is used i.e. a different classifier is used to combine the output of inner layer other than using any combination method. The results indicate that this heterogeneous approach is better in terms of effectiveness, stability and efficiency [5] . Dario et al. [18] proposed a very unique model that is dynamic selection of classifiers. This model uses the structural characteristics of data to select which classifier gives best prediction result and to select the best classifier random forest model is used. Santosh et al. [19] proposed an heterogeneous ensembling approach for building an efficient fault prediction model. Different classifiers are used as base classifiers that make independent errors. Linear and non-linear combination rules are used to combine the outputs of the classifier. This model greatly helps the practitioners to identify faults with less effort. Shahid et al. performed a study to evaluate the performance of different ensemble methods i.e. boosting, voting and stacking. For this five base classifiers are used for ensembling and ck design metrics are used to find the relationship between faults. The study is performed on twelve datasets and the results show that ensembling has better prediction power than single classifier. [2015,40] Anurag et al. proposed a hybrid model for prediction of faults using machine learning, genetic programming and ABC algorithm. Features performed a significant role in predicting the faults in a module that why wrapper method is used to select the useful attributes. The result of this study proved that hybrid model has better prediction performance.

3. CRITICAL ANALYSIS OF EXISTING MODELS

After the detailed analysis of the recent fault prediction models based on heterogeneous ensembling, a critical analysis of is tabulated in table 1. A newly proposed model performs better than existing models and the limitations of that model will become the future scope for the new researcher.

Table 1. Critical analysis of recent fault prediction models based on heterogeneous ensembling

References	Focused Area	Metrics proposed	Methods/Models/Techniques used	Technique predicted to be superior	Limitations/ Future Scope
1	Comparison between machine learning and Ensembling technique	LOC, Halstead , McCabe,CK	Ensembling and machine learning techniques	Ensembling	Only SMOTE is used for feature selection Many meta heuristic techniques can be used for attribute selection
2	SFP using Ensembling	• Class level metrics • Method level metrics	Ensembling Techniques • Bagging • Boosting • Stacking • Voting	Bagging	• This study is not generalized. • Default parameters of the tool is used. • No empirical evaluation • No feature selection technique is used.
3	SFP using Ensembling	• Halstead Metrics • Information Gain • SMOTE	• Combined Ensemble learning Approach	• Combined approach	• Enhanced model that deals with class imbalance and noise. • Can be replicated with different metrics and different feature selection techniques.
4	Evaluation of SDP model Using ANP	• CfsSubsetEval	• Machine learning Techniques • Ensembling techniques	• Decision tree • Boosted SMO,Voting, Stacking	• Accuracy is not the only way to access the performance of the model.
5	Early prediction of faults	• Product metrics • Process metrics	• Machine learning techniques • Statistical techniques	• Machine learning techniques	• Less no. of studies that actually work of early prediction of faults.
6	Ensemble model to handle class imbalance	• Specific features related to data	• RF • ROI • MWM	• Ensemble	• This model carries the variance problem of individual classifier
7	Ensemble model for SFP	• Source code metrics	• Heterogeneous ensemble • LOGR • ANN,BTE	• Heterogeneous ensemble with MVE	• Needs to be replicated to generalize the findings of the study.
8	Two layer approach for SFP	• Change Metrics	• Decision tree • Bagging • Random Forest • Stacking	• Stacking	• Needs more work with different ML to make it generalized.
9	Dynamic Selection of classifiers for SDP	• Product metrics	• ASCI • DT • VV	• ASCI	• This model can be implemented on cross project defect

					prediction.
10	Ensemble model for SDP	• Size metrics	• Homogeneous ensembling • Heterogeneous ensembling • LCR • NLCR	• Heterogeneous ensembling • NLCB	• More combination rule needs to be used to combine the outputs of the base classifier.
11	Ensemble model for software fault prediction	• CK metrics Suite	• AdaboostM1 • Vote • StackingC	• StackingC	• More study needs to be done to generalize the findings of the study
12	Hybrid model for SFP	• Procedural metrics • OOM	• SVM • ABC • Symbolic Regression	• Ensembling	• In addition to defects more quality attributes can be considered .

Omparative analysis of these models shows that the existing models are not fully optimized and there are still various issues that need to be considered. The proposed model considers these issues and performs prediction in an efficient manner.

4. ISSUES AND CHALLENGING

Generalization is one of the most important issue for fault prediction studies. Almost all the studies implemented are based on the open source software projects. That is the reason why software developers do not believe in such models. So, there is a need to implement these models on commercial and proprietary projects to improve generalization [20]. Privacy is another concern because that is the reason that why the implementation on commercial projects is not available. So, policies should be made regarding privacy so that companies feel comfortable to share their private data for implementation [21, 22, 23]. Class imbalancing is the issue that needs more consideration because it directly affects the performance of the defect prediction model. Although lots of studies have been implemented on this issue but it still needs further work [24, 25, 26]. Performance evaluation measures are not adequate and consistent which leads to inappropriate conclusion. So further work on this is also required. Diversity is the major concern when building ensemble model. There are many ways to achieve diversity but sometimes the highly diverse model also not able to improve the result so care should be made before selection of classifiers. [19,27] Selection of features highly affects the prediction performance. Lots of work has done on it. But it still needs reconsideration. [27,28,29] Selection of base classifier is the most challenging issue when building ensemble model. [19,30]

5. PROPOSED MODEL

In the proposed model , we focus on all these challenges and issues that are discussed above. Here, we implement diversity in two ways. First, by using different base classifier and second, by generating different training subset using random sampling. Feature selection technique is used to select those features that are highly associated with the prediction of faults. It affects the prediction performance of the model. The overall framework of the proposed model is given below.

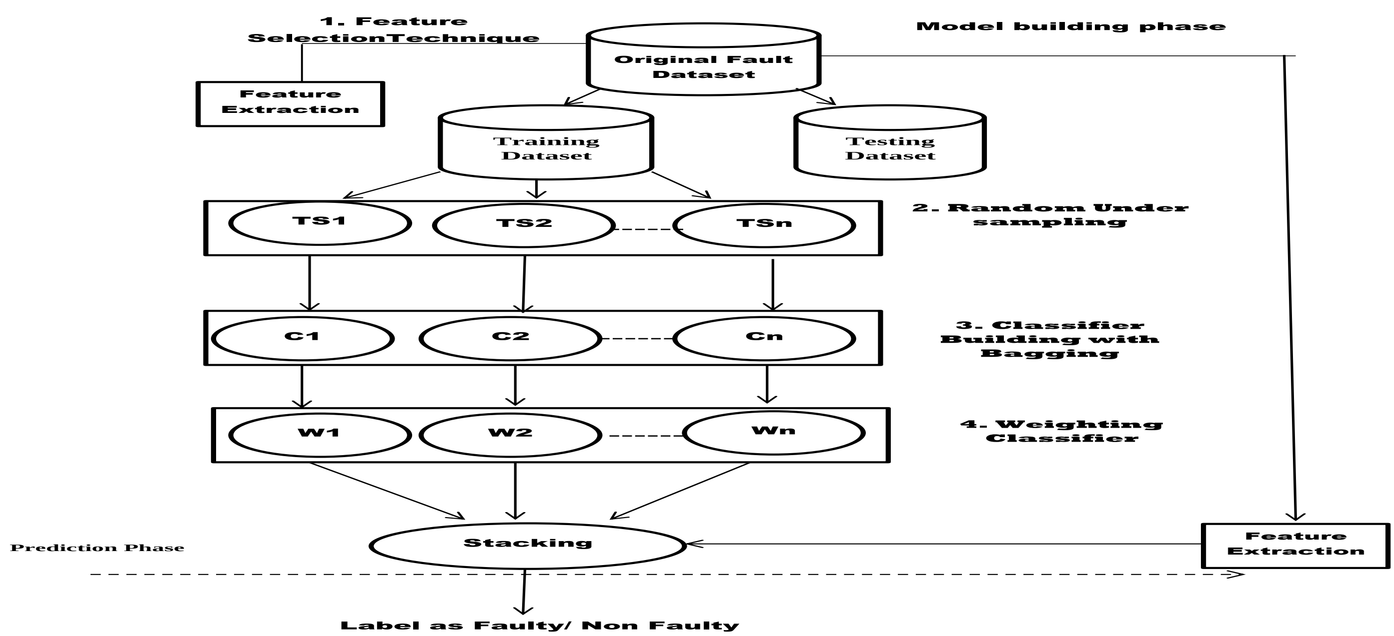

Fig. 1 The above figures presents the proposed model of fault prediction

In the proposed model we have two phase i.e. model building phase and prediction phase. In the model building phase, firstly we divide the whole dataset into two parts: training dataset and testing dataset. Then in order to achieve diversity, the training dataset is again divided into n number of datasets based on random under sampling technique. Later on, we train n classifiers based on different datasets. Different weights are assigned to different classifier based on the basis of their performance. In the prediction phase, we use stacking i.e. another classifier is used to predict the faults but that classifier is trained on the output of previous phase. The advantage of using it is that it will take the advantage of those classifiers that are able to detect unique faults.

CONCLUSION AND FUTURE SCOPE

Quality is the main concern of every software product. Fault prediction models contribute significantly in the improvement of quality. It also decreases the time, cost, and effort of the organization. There are various challenges that we still face when building an efficient fault prediction model. Therefore, we proposed a model that considers all these issues and helps the managers significantly in managing their resources. This is not implemented yet but in future we are planning to implement that model in various datasets and projects. In future, the presented model can be implemented on commercial projects to generalize the findings of the study. Also, different techniques can be used with stacking to get better prediction result.

REFRENCES

[1] Menzies, T., Milton, Z., Turhan, B., Cukic, B., Jiang, Y., & Bener, A. (2010). Defect prediction from static code features: current results, limitations, new approaches. Automated Software Engineering, 17(4), 375-407.

[2] Xu, Z., Liu, J., Yang, Z., An, G., & Jia, X. (2016, October). The impact of feature selection on defect prediction performance: An empirical comparison. In 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)(pp. 309-320). IEEE.

[3] Shatnawi, R., & Li, W. (2016). An empirical investigation of predicting fault count, fix cost and effort using software metrics. IJACSA) Int J Adv Comput Sci Appl, 7(2)..

[4] Giacinto, G., & Roli, F. (2001). Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognition, 34(9), 1879-1881.

[5] Yang, X., Lo, D., Xia, X., & Sun, J. (2017). TLEL: A two-layer ensemble learning approach for just-in-time defect prediction. Information and Software Technology, 87, 206-220.

[6] Bowes, D., Hall, T., & Petrić, J. (2018). Software defect prediction: do different classifiers find the same defects?. Software Quality Journal, 26(2), 525-552.

[7] Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.

[8] Rathore, S. S., & Kumar, S. (2017). Linear and non-linear heterogeneous ensemble methods to predict the number of faults in software systems. Knowledge-Based Systems, 119, 232-256.

[9] Ting, K. M., & Witten, I. H. (1997). Stacked Generalization: when does it work?.

[10] Healey, S. P., Cohen, W. B., Yang, Z., Brewer, C. K., Brooks, E. B., Gorelick, N., ... & Loveland, T. R. (2018). Mapping forest change using stacked generalization: An ensemble approach. Remote Sensing of Environment, 204, 717-728.

[11] Alsaeedi, A., & Khan, M. Z. (2019). Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study. Journal of Software Engineering and Applications, 12, 85-100.

[12] Shanthini, A. (2014). Effect of ensemble methods for software fault prediction at various metrics level.

[13] Yohannese, C. W., Li, T., & Bashir, K. (2018). A Three-Stage Based Ensemble Learning for Improved Software Fault Prediction: An Empirical Comparative Study. International Journal of Computational Intelligence Systems, 11(1), 1229-1247.

[14] Balogun, A. O., Bajeh, A. O., Orie, V. A., & Yusuf-Asaju, W. A. (2018). Software Defect Prediction Using Ensemble Learning: An ANP Based Evaluation Method.

[15] Abdou, A. S., & Darwish, N. R. (2018). Early Prediction of Software Defect using Ensemble Learning: A Comparative Study. International Journal of Computer Applications, 975, 8887.

[16] Huda, S., Liu, K., Abdelrazek, M., Ibrahim, A., Alyahya, S., Al-Dossari, H., & Ahmad, S. (2018). An ensemble oversampling model for class imbalance problem in software defect prediction. IEEE access, 6, 24184-24195.

[17] Kumar, L., Rath, S., & Sureka, A. (2017, July). An empirical analysis on effective fault prediction model developed using ensemble methods. In 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC) (Vol. 1, pp. 244-249). IEEE.

[18] Di Nucci, D., Palomba, F., Oliveto, R., & De Lucia, A. (2017). Dynamic selection of classifiers in bug prediction: An adaptive method. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(3), 202-212.

[19] Rathore, S. S., & Kumar, S. (2017). Towards an ensemble based system for predicting the number of software faults. Expert Systems with Applications, 82, 357-382.

[20] Tantithamthavorn, C., & Hassan, A. E. (2018, May). An experience report on defect modelling in practice: Pitfalls and challenges. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (pp. 286-295). ACM.

[21] Peters, F., Menzies, T., Gong, L., et al.: 'Balancing privacy and utility in cross-company defect prediction', IEEE Trans. Softw. Eng., 2013, 39, (8), pp.1054–1068

[22] Peters, F., Menzies, T.: 'Privacy and utility for defect prediction: experiments with morph'. Proc.

34th Int. Conf. Software Engineering, 2012, pp. 189–199

[23] Peters, F., Menzies, T., Layman, L.: 'Lace2: better privacy-preserving data sharing for cross project defect prediction'. Proc. 37th Int. Conf. Software Engineering, 2015, pp. 801–811

[24] Tan, M., Tan, L., Dara, S., & Mayeux, C. (2015). Online Defect Prediction for Imbalanced Data. Proceedings - International Conference on Software Engineering, 2, 99–108. https://doi.org/10.1109/ICSE.2015.139

[25] Wang, S., Yao, X.: 'Using class imbalance learning for software defect prediction', IEEE Trans. Reliability., 2013, 62, (2), pp. 434–443

[26] Malhotra, R., Khanna, M.: 'An empirical study for software change prediction using imbalanced data', Empirical Software Eng., 2017, 22, (6), pp. 2806–2851

[27] Baba, N. M., Makhtar, M., Fadzli, S. A., & Awang, M. K. (2015). CURRENT ISSUES IN ENSEMBLE METHODS AND ITS APPLICATIONS. Journal of Theoretical & Applied Information Technology, 81(2).

[28] Agrawal, A., & Menzies, T. (2018, May). Is better data better than better data miners?: on the benefits of tuning smote for defect prediction. In Proceedings of the 40th International Conference on Software engineering (pp. 1050-1061). ACM.

[29] Afzal, W., & Torkar, R. (2016). Towards benchmarking feature subset selection methods for software fault prediction. In Computational intelligence and quantitative software engineering (pp. 33-58). Springer, Cham.

[30] Kumar, L., Misra, S., & Rath, S. K. (2017). An empirical analysis of the effectiveness of software metrics and fault prediction model for identifying faulty classes. Computer Standards & Interfaces, 53, 1-32.