Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.11: 2024
ISSN :**1906-9685**

REVOLUTIONIZING SOFTWARE QUALITY: A NOVEL MACHINE LEARNING METHOD FOR DEFECT ESTIMATION

Mr.K Ravi Raju Assisstant Professor in Department of CSE, Raghu Institute Of Technology, Vishakapatnam.
Sumala Harika, Sasanapuri Mahidhar, Shaik Sharukh Khan, Pinisetty Rahul Varun, B.Tech with Specialization of Computer Science and Engineering in Raghu Institute Of Technology, Vishakapatnam.

Abstract_ Recent improvements in software defect prediction (SDP) involve the combination of different classification algorithms to form an ensemble or hybrid approach. This technique was developed to increase prediction performance by overcoming the constraints of individual classification techniques. This study conducts a systematic literature review on the application of the ensemble learning approach to software defect prediction. The review is performed after thoroughly reviewing research papers released since 2012 in four well-known online libraries: ACM, IEEE, SpringerLink, and ScienceDirect. This paper addresses five research problems related to the use of ensemble learning for software defect prediction. To extract the answers to the identified queries, 46 most relevant papers are picked after a rigorous systematic research procedure.In this research, we use ensemble machine learning algorithms including Random Forest, Logistic Regression, and Linear Regression to forecast software problems. Software flaws constantly present the issue of extending development time and investing additional money.

1.INTRODUCTION

The troupe learning model is worked by joining the numerous AI classifiers to further develop expectation execution [2]. As per the writing, many terms, like crossover, joined, coordinated, and collected arrangement, are utilized for gathering learning [20]-[23]. In the conventional strategy for imperfection expectation, a singular classifier, for example, the guileless Bayes classifier, choice trees, or a multi-facet perceptron, is utilized to fabricate the expectation model on a pre-named dataset. Individual classifiers could have a few shortcomings to foresee a specific deformity under a particular situation [24], [25]. Hence, group learning was applied with the goal that the qualities of various classifiers can be consolidated to give better deformity revelation in the dataset. Numerous scientists have given observational proof somewhat recently, which recommends that outfit strategies give preferred characterization exactness over individual classifiers [26]-[30]. Comprehensively, gathering techniques are arranged into two gatherings in view of the sorts of base students: 1) homogeneous troupe strategies and 2) heterogeneous group techniques. In homogenous gathering techniques, similar base students are applied to an alternate arrangement of examples in a dataset. Models incorporate sacking, helping, revolution woods, and so forth [18]. In the heterogeneous gathering strategy, different base students are created utilizing different AI procedures. These base students are joined, and last expectation is performed by incorporating the consequences of base students either genuinely or by casting a ballot [18]. Heterogeneous strategies are more assorted than homogeneous techniques because of the various qualities of base students. Group techniques can likewise be classified as straight and nonlinear. In straight troupe strategies, the result of base student models is consolidated utilizing a direct capability, like a weighted normal or straightforward normal, while in nonlinear group techniques, a nonlinear strategy, for example, a choice tree or backing vector machine (SVM), is applied to join the choice of base students [19]. Specialists additionally consider variety while adding different classifiers into a gathering. Variety of classifiers alludes to the idea that the picked classifiers in a group strategy commit errors about various occasions of information. Various measures are utilized to assess the variety between two classifiers, for example, the Relationship Variety Measure, Q-Insights, Accuracy, and Weighted Precision and Variety (Roll). Aside from headways in outfit learning methods, a lot seriously encouraging SDP approaches are being proposed. These methodologies expect to foresee deserts significantly sooner in the product improvement lifecycle utilizing the ideas of code scents and prerequisites smells. Hennning et al. [75] proposed a lightweight static necessities investigation approach named Smella that took into consideration prompt quick checks when prerequisites were down on paper.

2.LITERATURE SURVEY

2.1 S. Parnerkar, A. V. Jain, and C. Birchha, ‘‘An approach to efficient software bug prediction using regression analysis and neural networks,’’ Int. J. Innov. Res. Comput. Commun. Eng., vol. 3, no. 10, Oct. 2015.

Machine Learning approaches are good in solving problems that have less information. In most cases, the software domain problems characterize as a process of learning that depend on the various circumstances and changes accordingly. A predictive model is constructed by using machine learning approaches and classified them into defective and non-defective modules. Machine learning techniques help developers to retrieve useful information after the classification and enable them to analyse data from different perspectives. Machine learning techniques are proven to be useful in terms of software bug prediction. This study used public available data sets of software modules and provides comparative performance analysis of different machine learning techniques for software bug prediction. Results showed most of the machine learning methods performed well on software bug datasets. The advancement in software technology causes an increase in the number of software products, and their maintenance has become a challenging task. More than half of the life cycle cost for a software system includes maintenance activities.

2.2 B. Liu, H. Qin, Y. Gong, W. Ge, M. Xia, and L. Shi, ''EERA-ASR: An energy-efficient reconfigurable architecture for automatic speech recognition with hybrid DNN and approximate computing,'' IEEE Access, vol. 6, pp. 52227–52237, 2018.

This paper proposes a hybrid deep neural network (DNN) for automatic speech recognition and an energy-efficient reconfigurable architecture with approximate computing for accelerating the DNN. To accelerate the hybrid DNN and reduce the energy consumption, we propose a digital–analog mixed reconfigurable architecture with approximate computing units, including a binary weight network accelerator with analog multi-chain delay-addition units for bit-wise approximate computing and a recurrent neural network accelerator with approximate multiplication units for different calculation accuracy requirements. Implemented under TSMC 28nm HPC+ process technology, the proposed architecture can achieve the energy efficiency of 163.8TOPS/W for 20 keywords recognition and 3.3TOPS/W for common speech recognition. Deep Neural Networks (DNNs) that have many hidden layers have been proven to outperform traditional models (i.e., Markov models, Gaussian mixture models) on a variety of speech recognition benchmarks by a large margin [1], [2].

2.3 N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. W. Schuller, ''An image-based deep spectrum feature representation for the recognition of emotional speech,'' in Proc. 25th ACM Multimedia Conf. (MM), 2017, pp. 478–484.

The outputs of the higher layers of deep pre-trained convolutional neural networks (CNNs) have consistently been shown to provide a rich representation of an image for use in recognition tasks. This study explores the suitability of such an approach for speech-based emotion recognition tasks. First, we detail a new acoustic feature representation, denoted as deep spectrum features, derived from feeding spectrograms through a very deep image classification CNN and forming a feature vector from the activations of the last fully connected layer. We then compare the performance of our novel features with standardised brute-force and bag-of-audio-words (BoAW) acoustic feature representations for 2- and 5-class speech-based emotion recognition in clean, noisy and denoised conditions. The presented results show that image-based approaches are a promising avenue of research for speech-based recognition tasks. Key results indicate that deep-spectrum features are comparable in performance with the other tested acoustic feature representations in matched for noise type train-test conditions; however, the BoAW paradigm is better suited to cross-noise-type train-test conditions. Convolutional neural networks (CNNs) have become increasingly popular in machine learning research.

4.PROPOSED SYSTEM

In the proposed approach, ensemble learning is used to combine the strengths of different classifiers to improve defect detection in the dataset. Over the past decade, numerous studies have found that ensemble approaches outperform individual classifiers in terms of accuracy..

4.1 ALGORITHM DETAILS

SVM Algorithm: Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyper plane which separates the data into classes. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. As a simple example, for a classification task with only two features (like the image above), you can think of a hyper plane as a line that linearly separates and classifies a set of data. Intuitively, the further from the hyper plane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyper plane as possible, while still being on the correct side of it.

So when new testing data is added, whatever side of the hyper plane it lands will decide the class that we assign to it.

Random Forest Algorithm: it’s an ensemble algorithm which means internally it will use multiple classifier algorithms to build accurate classifier model. Internally this algorithm will use decision tree algorithm to generate it train model for classification.

Bagging: This algorithms work similar to learning tree the only difference is voting concept where each class will get majority of votes based on values close to it and that class will form a branch. If new values arrived then that new value will applied on entire tree to get close matching class.

KNN: K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

.

4.RESULTS AND DISCUSSION

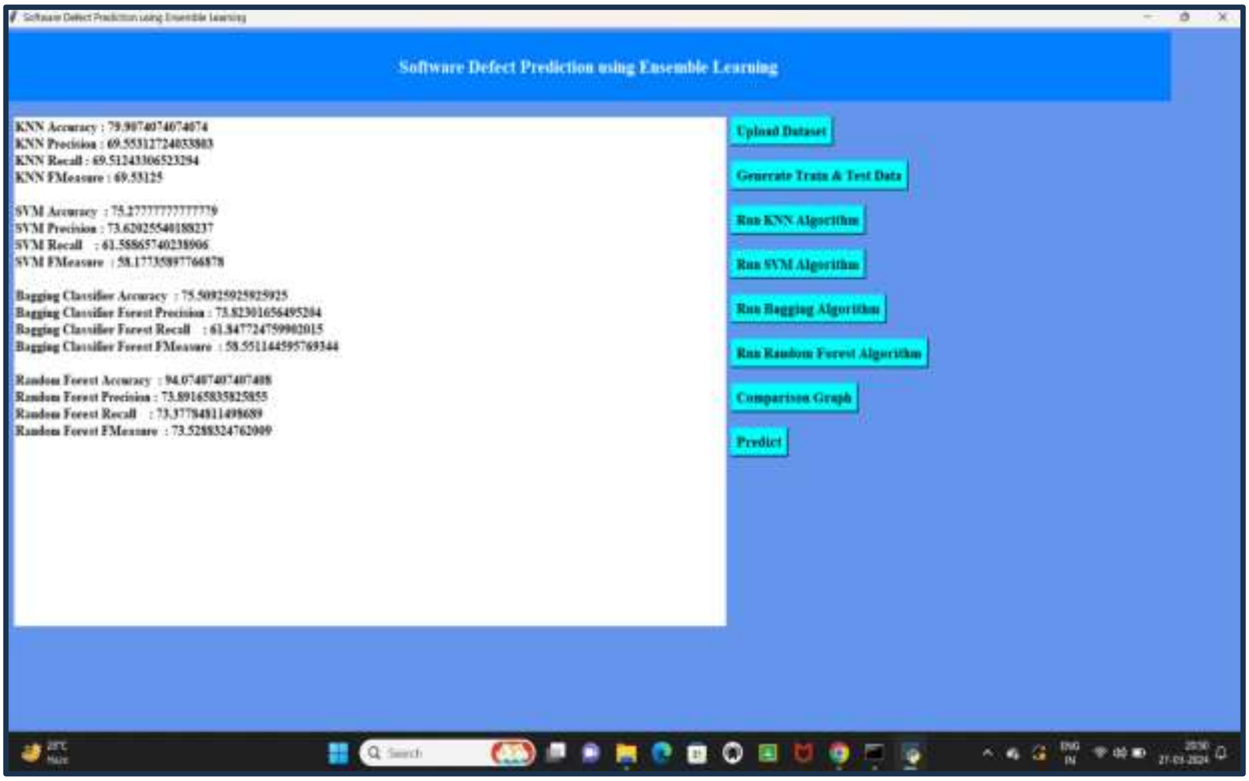

Fig 1: Performing Algorithms on the data:

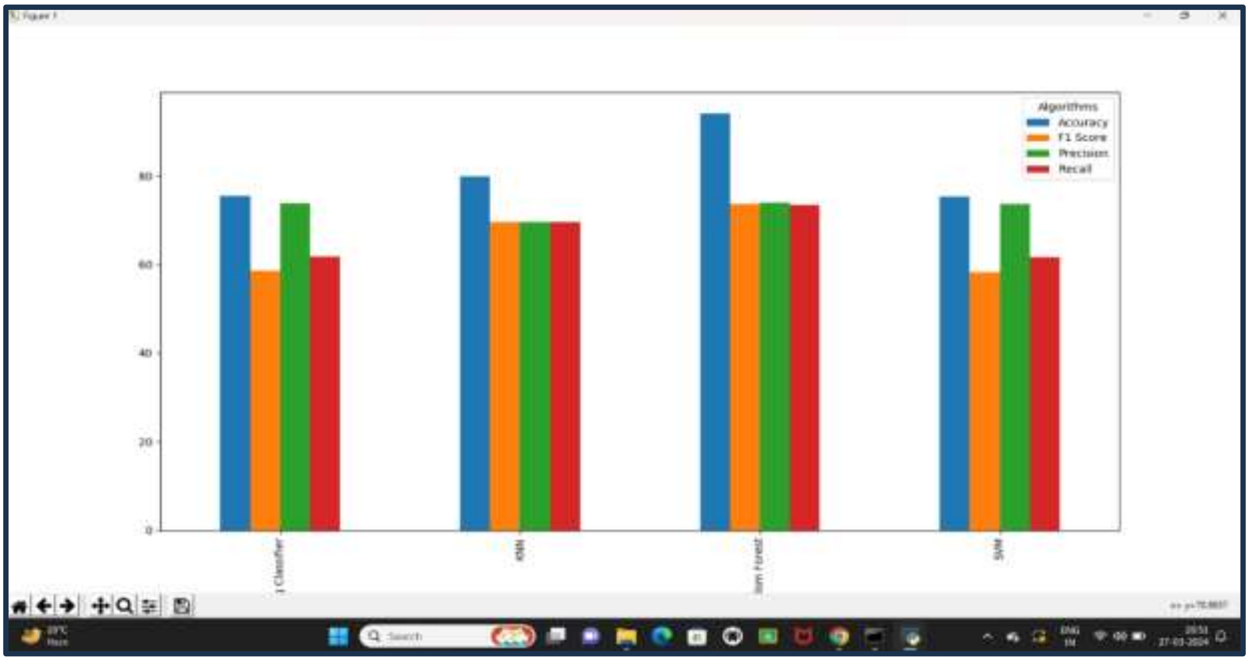

Fig 2:Graphical representation of the prediction:

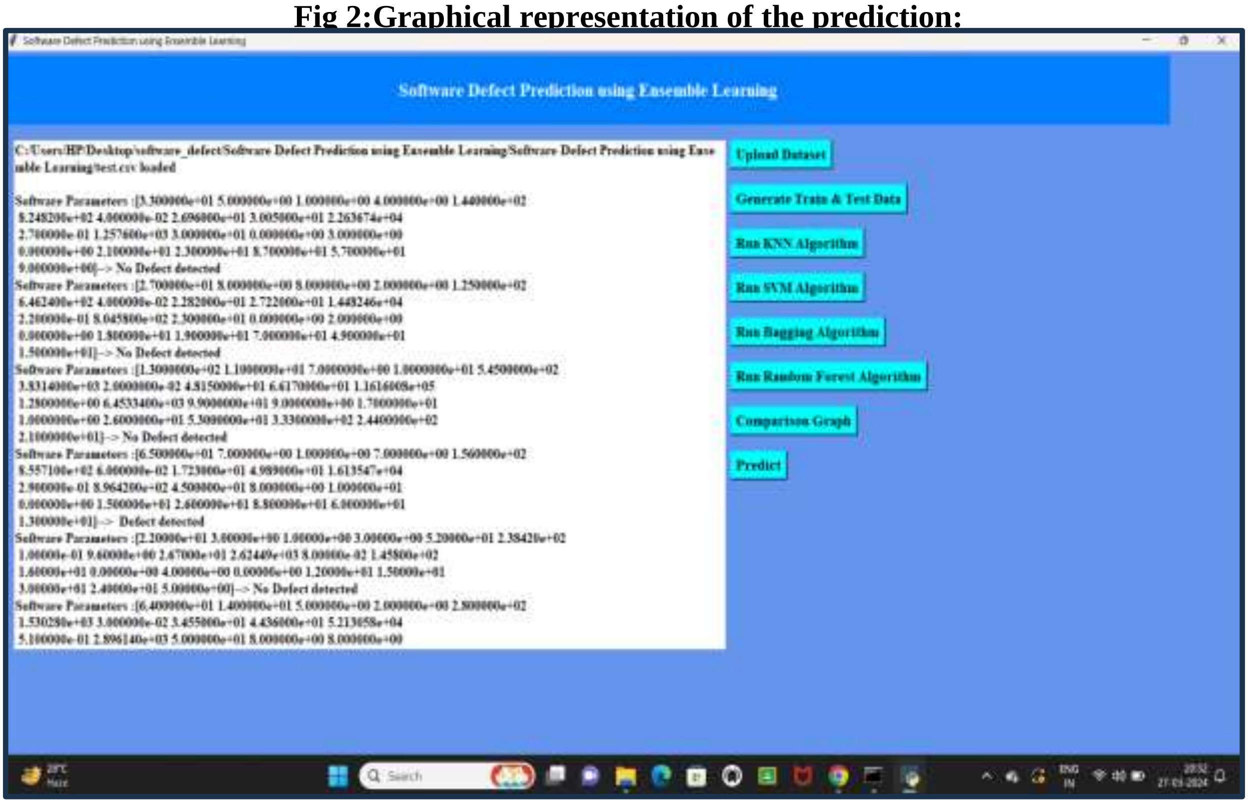

Fig 3:Prediction Screenshot

5.CONCLUSION

This work uses an SLR to follow the most current research developments in ensemble learning techniques for software defect prediction. This review is conducted by rigorously assessing the most important research papers published in three well-known online libraries: ACM, IEEE, Springer Link, and Science Direct. This study defines and addresses five research topics related to the various aspects of research advancement on the application of ensemble learning techniques for software defect prediction. It is concluded that ensemble learning strategies outperformed individual classifiers. In the future, the implications of feature selection approaches on ensemble learning should be reviewed.

.

REFERENCES

1.J. Petrić, D. Bowes, T. Hall, B. Christianson and N. Baddoo, "Building an ensemble for software defect prediction based on diversity selection", Proc. 10th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas., pp. 1-10, Sep. 2016.

Show in Context CrossRef Google Scholar

2.A. N. R. Moparthi and B. D. N. Geethanjali, "Design and implementation of hybrid phase based ensemble technique for defect discovery using SDLC software metrics", Proc. 2nd Int. Conf. Adv. Electr. Electron. Inf. Commun. Bio-Inform. (AEEICB), pp. 268-274, Feb. 2016.

Show in Context View Article

Google Scholar

3.S. S. Rathore and S. Kumar, "Ensemble methods for the prediction of number of faults: A study on eclipse project", Proc. 11th Int. Conf. Ind. Inf. Syst., pp. 540-545, Dec. 2016.

Show in Context View Article

Google Scholar.

[4] Norman Fenton, Paul Krause, and Martin Neil. Software measurement: Uncertainty and causal modeling. IEEE software, 19(4):116–122, 2002.

[5] Lan Guo, Yan Ma, Bojan Cukic, and Harshinder Singh. Robust prediction of fault-proneness by random forests. In 15th International Symposium on Software Reliability Engineering, pages 417–428. IEEE, 2004.

[6] Taghi M Khoshgoftaar, Edward B Allen, and Jianyu Deng. Using regression trees to classify fault-prone software modules. IEEE

Transactions on reliability, 51(4):455–462, 2002.

[7] Taghi M Khoshgoftaar, Edward B Allen, John P Hudepohl, and Stephen J Aud. Application of neural networks to software quality modeling of a very large telecommunications system. IEEE Transactions on Neural Networks, 8(4):902–909, 1997.

[8] Sunghun Kim, Hongyu Zhang, Rongxin Wu, and Liang Gong. Dealing with noise in defect prediction. In 2011 33rd International Conference on Software Engineering (ICSE), pages 481–490. IEEE, 2011.

[9] Yan Ma, Lan Guo, and Bojan Cukic. A statistical framework for the prediction of fault-proneness. In Advances in Machine Learning Applications in Software Engineering, pages 237–263. IGI Global, 2007.

[10] Ruchika Malhotra. A systematic review of machine learning techniques for software fault prediction. Applied Soft Computing, 27:504–518, 2015.

[11] Jinsheng Ren, Ke Qin, Ying Ma, and Guangchun Luo. On software defect prediction using machine learning. Journal of Applied Mathematics, 2014, 2014.

[12] J. Sayyad Shirabad and T.J. Menzies. The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada, 2005.

[13] Shuo Wang and Xin Yao. Using class imbalance learning for software defect prediction. IEEE Transactions on Reliability, 62(2):434–443, 2013.