A NOVEL XGBOOST TUNED MACHINE LEARNING MODEL FOR SOFTWARE BUG PREDICTION

G. Giridhar Naidu
PG Scholar
Department of Computer Applications
Madanapalle Institute of Technology & Science, Madanapalle, Chittoor Dist, AP, India

Mr. S. Balamurugan
Assistant Professor
School of Computers
Madanapalle Institute of Technology & Science, Madanapalle, Chittoor Dist, AP, India

Mrs. S. Savitha
Assistant Professor
Department of Computer Science
Saradha Gangadharan College, Pondicherry

***Abstract*: Software bug prediction is important during software development and maintenance. The early prediction of defective modules in developing software can help the development team to utilize the available resources efficiently and effectively to deliver high quality software product in limited time. Machine learning approach works by extracting the hidden patterns among software attributes. In this study, several machine learning classification techniques are used to predict the software defects in NASA datasets JM1, CM1, KC2 and PC3.It was proposed based on tuning the existing XGBoost model. The results achieved were compared model outperformed them for all datasets.**

***Keywords*: Machine Learning, Dataset, Supervised Learning, Random Forest, XG Boost, Ada Boost, Decision Tree.**

1. INTRODUCTION:

Nowadays, completing a software project successfully is a major challenge [1]. Defects or bugs are a project manager's worst nightmare. These bugs occur as a result of poor code design and implementation. The level of knowledge is the most difficult challenge in writing defect-free code [2]. Consider a team of 5-6 developers working on a project, some of whom are experienced and others who are newer [3]. Now, the new developer has little experience with the types of defects that can occur in this code in real-world scenarios. As a result, they simply implement the project without regard for future bugs. After that application is distributed to users, they will encounter bugs in a non-uniform environment, affecting the application rating and customer satisfaction. Industry is to develop a 100% bug free application. This issue is difficult to achieve by the software development companies even if they are kept on testing [6]. Basically, any application developed by human is not an automated process so having defect is a common or natural thing Nevertheless, the software development companies focus on early defect detection though several inspections, testing procedures. So, to resolve this issue we reviewed a various approach based on machine learning [7].

2. LITERATURE SURVEY:

He compared most machine learning approaches, including both supervised and unsupervised learning, in his Methodology. WEKA Tool was used for the experiment, as well as PROMISE -NASA. The model is trained using a data set [8]. For accurate detection, a retrieval and classification model based on (CNN) and Long Short-Term Memory (LSTM) was introduced [9]. A method was proposed using a historical data set and the Supervised Learning algorithm, primarily logistic regression, Nave Bayes, and Decision Tree. And the KFold cross validation technique [10] was used. Outlier detection and removal were prioritised, followed by dimension reduction [11] Proposed bug detection as a binary classification problem, e.g., correct and incorrect, trained the classifier to differentiate between incorrect and correct code using the deep Bugs framework [12]. A defect detector framework is proposed as a tool or framework that works with various compilers and languages such as javac, gcc, and visual studio. [13]. By using marginal R square values, an approach is proposed that uses the fewest and most accurate number of performing metrics at the same time. Eclipse JDT Core dataset is used. [14]. One Class SVM was used to propose a one-class SFP (Software Fault Prediction) Model. [15]. Machine learning was used to predict vulnerability in a web application. This paper generates input validation and sanitation attributes. For each sink, it computes a static backward slice. The programme analysis is founded on Approach suggested is first classify the bugs based on their priorities based on severity and component attribute. Uses Mean Clustering algorithm with Bayes Net Classifier [17]. Uses Supervised learning. Datasets Used: KC1, MC1, AR1, AC6, MC2 to train the model then compares the results of naïve Bayes and j48(Decision Tree Classifier) [18]. Used Supervised learning on 10 Data Sets Provided by means of NASA especially classifiers used are Bagging, guide vector machines (SVM), choice tree (DS), and random wooded area (RF) classifiers [19]. Data is collected from an open Source Software where data will be in a form of objectoriented matrices. Model proposed is genetic based Classifier Systems [20].

3. PROPOSED SYSTEM:

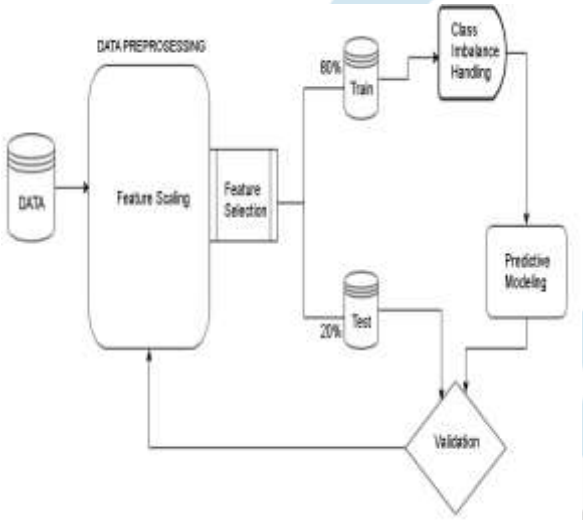
In our examination and broad writing review, we tracked down that Random Forest turns out great for Software Bug Prediction an extraordinary precision yet it tends to be additionally expanded by other recently show up calculation adaptable start to finish boosting framework called as XG Boost in this exploration we are proposing a strategy to assemble an arrangement model with more prominent exactness than Random Forest.

ADVANTAGES:

- Higher Accuracy.
- Low fluctuation in characterization.
- Bias because of presumption about dataset are least or even nonexistent.
- High execution speed.

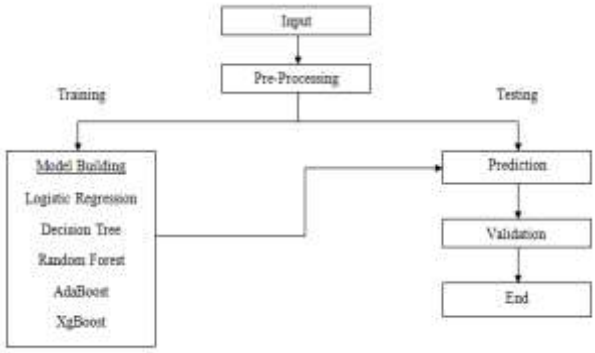
4. SYSTEM DESIGN

Methodology: In our research and extensive literature review, we unearthed that Random Forest works well for Software Bug Prediction with a high accuracy, but it can be further improved by yet another newly introduced optimization method XGBoost. In this research project, we recommend an approach is to construct a predictive model with a maximum reliability than Random Forest.



Fig(I) Flowchart presenting execution Process

System Architecture



Fig(II) System Architecture

In our research and extensive literature review, we discovered that Random Forest works well for Software Bug Prediction with a high accuracy, but it can be further improved by another recently introduced algorithm called XGBoost. In this study, we propose a method for building a predictive model with a higher accuracy than Random Forest.

In the above Process diagram, it shows a detailed explanation how our Project takes the Input and processes that and gives the Results. Here we explain it in the stepwise format,

In order to give Input to the System/Project we have to gather data fromdifferent repositories. One among the repositories that we used is "Pedestrian Detection", where it gives the human related Data like (human images in different locations).
The Mobile Net model was proposed by Google and is a type of base architecture highly suitable for embedded-based vision applications with lesscomputing power. The Mobile Net architecture uses depth wise separable convolutions instead of standard convolution. This reduces the number of parameters significantlyas compared to the network with normal convolution with the same amount of depth inthe network, which results in lightweight deep neural networks. The activation function "Re LU " is replaced by "ReLU6", and the "Batch Normalization" layer.

In the above Process diagram, it shows a detailed explanation how our Project takes the Input and processes that and gives the Results. Here we explain it in the stepwise format.

In order to give Input to the System/Project we have to gather data fromdifferent repositories. One among the repositories that we used is "Pedestrian Detection", where it gives the human related Data like (human images in different locations).

ADVANTAGES:

- Reduce ubnormal acitivities
- To reduce cases of various viruses.
- To maintain safety regulations.

Data Prepossessing

Machine learning is the most fascinating thing going on right now. Everyone is beginning to use machine learning models in their businesses. Data is at the heart of the complex process. Our machine learning tools are effective in terms of data quality. As a result, data pre-processing is an important step in the development of an effective prediction model. It is a method for determining the range of independent variables or data characteristics. In our study, we used the normalisation technique to convert our data to the same scale.

Software Defect Dataset

In order to find the effectiveness of our approach, we choose a standard software bug dataset of NASA Promise Repository. From which KC2, PC3, JM1, CM1 datasets are used. This repository is publicly available at http://promise.site.uottawa.ca. Each dataset contain data of 22 excluding PC3 which contains 38 attributes respectively. The concise description of these dataset is as follows- The NASA dataset was developed under its Metrices Data Program. In 2013, Shepperd et al. have cleaned up duplicate and incompatible data from the dataset. This enhanced information can be found in the PROMISE database. So, in our study, we used NASA Dataset for the purified data. For NASA data, they use Halstead and McCabe metrics for each occurrence. There are approximately 40 features, including unique operator value (MU1), unique operator value (MU2), the total number of operators (N1), the total number of operands (N2), rows Code (LOC), etc. NASA project, each project uses different features.

Dataset	Project name	# Instances	# Defects	% defects	features
NASA	JM1	7782	1672	21.50%	22
	CM1	498	49	9.83%	22
	KC2	522	105	20.50%	22
	PC3	1077	134	12.40%	38

Logistic Regression:

Logistic Regression is a machine learning classifier that is supervised. It predicts the outcome in the form of a class, such as (0 or 1) or True or False. It is a subset of linear regression, but linear regression predicts in a continuous fashion, whereas logistic regression predicts in a binary fashion. Logistic Regression is used in our scenario to predict whether or not the code is defective. It allows us to determine how the specific effect of a variable on a prediction differs from 0 if not, indicating that the variable is not useful for prediction and we can remove it from the model.

Decision Tree

- Decision Tree is a Supervised learning technique that can be used for both classification and regression problems, but it is most commonly used for classification. It is a tree-structured classifier, with internal nodes representing dataset features, branches representing decision rules, and each leaf node representing the result. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

XGBoost:

Extreme Gradient Boosting (XG Boost) is an open-source library that implements the gradient boosting algorithm in an efficient and effective manner. This tutorial will teach you how to create and test XG Boost regression models in Python. XG Boost is a fast gradient boosting implementation that can be used for regression predictive modelling.XGBoost Performed very well on our model we got Accuracy of 93% without tuning and after we performed tuning of Estimator, learning rate, max depth, subsample parameters from where we got a great accuracy enhancement and got a increment of 4% in accuracy with
97% of Accuracy, 3% increase in Precision with 95% precision, 4% increase in Recall with 98% of Recall and 3% increase in AUC with 96%. The results achieved using XG
Boost are shown in Table

Performance Parameter	Without Tuning	After Tuning
Accuracy	93.25%	97%
Precision	92.25%	95%
Recall	94.0%	98%
AUC	93.25%	96.75%

5. RESULTS:

We compared all of the models in order to validate and compare the results of other approaches to the one we presented. We tested five algorithms in this experiment: logistic regression, Decision Tree, Random Forest, ADA Boost, and Tuned XGBoost. Tuned XGBoost is the proposed model, in which we tweaked the N estimator, learning rate, max depth, and subsample parameters against four datasets: KC2, CM1, PC3, and JM1. The outcomes for cutting-edge models.The results achieved for state-of-art models is compared with our model

For PC3 dataset, our proposed model with Tuned XG Boost resulted into better classification accuracy of accuracy, precision, recall and AUC of 0.97, 0.95, 0.96, and 0.98 in comparison to other classifiers in state-of-art.

The classification report for the CM1 dataset indicates that our proposed model of Tuned XG Boost resulted in better classification accuracy of 0.95, 0.97, 0.95, and 0.96 in comparison to other state-of-the-art classifiers. In our model, we tuned an existing XGBoost model by changing one of its parameters, namely (Estimator, learning rate, max depth, and subsample).

6. CONCLUSION:

We tend to pre-prepare the information for this investigation by including scaling for higher extraction and decision. Using the SMOTE procedure, we addressed the issue of refinement unevenness in datasets. Then, for four datasets from NASA-KC2, PC3,

JM1, and CM1, we used AI models – calculated relapse, call Tree, Random Forest, adenosine deaminase lift, and XG Boost as condition of-craftsmanship models. Later, a new model was proposed that supported normalisation of the dominant XG Boost model by explicitly defining its boundary NEstimator, learning rate, max profundity, and subsample. The results obtained were compared to condition of workmanship models, and our model outperformed them for all datasets. The creators agree that this investigation can help in properly characterising bugs using an AI approach.

REFERENCES:

[1] Saiqa Aleem, Luiz Fernando Capretz and Faheem Ahmed "Benchmarking machine learning technique for software defect detection" IJSEA, Vol.6, No.3, May 2015.

[2] Jayati Deshmukh, Annervaz K M, Sanjay Podder, Shubhashis Sengupta, Neville Dubash "Towards Accurate Duplicate Bug Retrieval using Deep Learning Techniques"2017 IEEE.

[3] S. Delphine Immaculate, M. Farida Begam, M. Floramary "Software Bug Prediction Using Supervised Machine Learning Algorithms" 2019 IEEE.

[4] Surbhi Parnerkar, Ati Jain, Vijay Birchha "An Approach to Efficient Software Bug Prediction using Regression Analysis and Neural Networks" IJIRCCE 2015.

[5] Rashid, Mamoon, Lovepreet Kaur "Finding Bugs in Android Application using Genetic Algorithm and Apriori Algorithm." Indian Journal of Science and Technology 9.23 (2016): 1-5.

[6] Markland J. Benson "Toward Intelligent Software Defect Detection" NASA: 2019.

[7] Shruthi Puranika, Pranav Deshpandea, K Chandrasekarana "A Novel Machine Learning Approach for Bug Prediction" ScienceDirect, 2016.

[8] LIN CHEN, BIN FANG, ZHAOWEI SHANG "Software fault prediction based on One-Class SVM" IEEE -2016.

[9] Vignesh M, Dr. K. Kumar "Web Application Velnerability prediction using machine learning" IJSER, 2017.

[10] Neetu Goyal, Naveen Aggarwal, and Maitreyee Dutta "A Novel Way of Assigning Software Bug Priority Using Supervised Classification on Clustered Bugs Data" Springer, 2015.

[11] Meenakshi, Dr. Satwinder Singh "Software Bug Prediction using Machine Learning Approach" IRJET, 2019.

[12] Abdullah Alsaeedi, Mohammad, Zubair Khan "Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study" JSEA, 2019.

[13] Amod Kumar, Ashwni Bansal "Software Fault Proneness Prediction Using Genetic Based Machine Learning Techniques" IEEE,2019.

[14] Meiliana, Syaeful Karim, Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol, Edi Abdurachman, Benfano Soewito "Software Metrics for Fault Prediction Using Machine Learning Approaches" IEEE-2017.

[15] Keita Mori and Osamu Mizuno "An Implementation of Just-In-Time Fault-Prone Prediction Technique Using Text Classifier" IEEE, 2015.

[16] Ali Ouni, Marwa Daagi, Marouane Kessentini, Salah Bouktif, Mohamed Mohsen Gammoudi. "A Machine Learning-Based Approach to Detect Web Service Design Defects" IEEE, 2017.

[17] Uma Subbiah, Muthu Ramachandran and Zaigham Mahmood "Software Engineering Approach to Bug Prediction Models using Machine Learning as a Service (MLaaS)" IEEE-2019.

[18] Ashima Kukkar, Rajni Mohana, Anand Nayyar, Jeamin Kim, Byeong-Gwon Kang and Naveen Chilamkurti "A Novel Deep-Learning Based Bug Severity Classification Using Convolutional Neural Networks and Random Forest with Boosting" IEEE-2019.

[19] Awni Hammouri, Mustafa Hammad, Mohammad Alnabhan, Fatima Alsarayrah "Software Bug Prediction using Machine Learning Approach" Research Gate 2018.

[20] Fei Wu, Xiao-Yuan Jing, Ying Sun, Jing Sun, Lin Huang, Fangyi Cui, and Yanfei Sun "Cross-Project and Within-Project Semi supervised Software Defect Prediction: A Unified Approach" IEEE,2018.

[21] M. K. Rafsanjani, H. Bagherinezhad and R. S. Batth, "The Classification of Galaxy Images Using Neural Network Algorithm," *2019 International Conference on Computing, Power and Communication Technologies (GUCON)*, NCR New Delhi, India, 2019, pp. 127-131.