iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IX **Month of publication:** September 2024

DOI: https://doi.org/10.22214/ijraset.2024.64162

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IX Sep 2024- Available at www.ijraset.com

A Framework for Developing Correct Software Programs through Software Defect Prediction and Elimination using Machine Learning Models

Madhab Paul Choudhury[1], J Paul Choudhury[2]
[1]*PhD Research Scholar, IIT ISM Dhanbad, Jharkhand*
[2]*Retired Professor (IT), Kalyani Govt Engg College, Kalyani, WB, Professor(CSE) Narula Institute of Technology Kolkata*

Abstract: ***Software Defect Prediction [SDP] is an important item for development of mistake free software. A software defect is an error, bug, flaw, fault, malfunction or mistakes. If software defect is present in the software, the output produced from that software becomes erroneous. If defect is present in the software, time, cost, effort will be lost and there will be wastage of resources. Therefore, it is necessary is to determine the defects in an early phase of software development.***
For this purpose, Kaggle software defect data set(ant-1.3) [11] have been used. Machine learning models like Random Forest, Decision Tree, Logistic Regression, K-nearest neighbour, Gaussian Naïve Bayes, Support Vector Machine using linear function and Radial basis function, Gradient Boosting algorithm have been used. Accuracy, classification report and confidence matrix have been used as an evaluation parameter for selecting a particular machine learning model. After the selection of machine learning model based on accuracy, classification report and confidence matrix, the particular machine learning model has to be selected. It is necessary to find out the contribution of software parameters which are responsible for eliminating fault from software in a better way in the computer software.
Keywords: Machine Learning models; Random Forest; Decision Tree; Logistic Regression; K-nearest neighbour algorithm;

I. INTRODUCTION.

Software Defect Prediction [SDP] is an important item for development of mistake free software. A software defect is an error, bug, flaw, fault, malfunction or mistakes. If software defect is present in the software, the output produced from that software becomes erroneous. If defect is present in the software, time, cost, effort will be lost and there will be wastage of resources. Therefore, it is necessary is to determine the defects in an early phase of software development.

II. LITERATURE REVIEW

The authors [1] have selected seven distinct algorithms from machine learning techniques and are going to test them using the data sets acquired for NASA public promise repositories. The models used are SVM, Multilayer Perceptron, Run Bagging algorithm, Naive Bayes algorithm, Random Forest algorithm, Multinomial NB and Radial Basis Functions. The results of those models are satisfactory and the users can omit the error from that paper. To improve the quality of software, datamining techniques [2] have been used to find out predictions regarding the failure of software components forming software defects. Datamining techniques have been used. The authors have used Supervised Learning model and Un Supervised Learning models. Supervised learning models include Decision tree classification algorithm, Support vector machine (SVM), k-Nearest Neighbours, Naive Bayes, Random forest, Neural networks, Polynomial regression and SVM for regression. Regression models used are Linear Regression, Logistic Regression, Polynomial Regression, Lasso Regression and Multivariate Regression. Unsupervised learning models include K – Means clustering, Hierarchical clustering, Make Density Based Clustering. The authors [3] have tried to analyse the machine learning algorithms' performance for software defect classification. They have used seven datasets available from the NASA promise dataset repository here. The performance of Neural Networks and Gradient Boosting classifier have given better performance of other algorithms. A cost-sensitive deep ladder network-based software defect prediction model [4] has been proposed, which minimises the negative impact of the class imbalance problem on defect prediction. To eliminate the problem of lack or insufficiency of historical data, a flow learning-based geodesic cross-project software defect prediction method has been proposed. An improved deep belief network approach has been proposed for real-time defect prediction. The authors have shown that the defect prediction model learned by the improved method has shown better prediction performance.

The main aim of this paper [5] is to evaluate the capability of machine learning algorithms in software defect prediction and find the best category while comparing seven machine learning algorithms within the context of four NASA datasets obtained from public PROMISE repository. Machine learning algorithms based on Ensemble Learners, Bayesian Learners, Neural Networks and SVM. Here seven different machine learning algorithms to estimate software defect.

A software bug prediction model based on machine learning (ML) algorithms has been proposed [6]. Three supervised ML algorithms have been used to predict future software faults based on historical data. These classifiers used are Naïve Bayes (NB), Decision Tree (DT) and Artificial Neural Networks (ANNs). The evaluation process has shown that ML algorithms have high accuracy rate.

In this paper[7], a machine learning approach has been proposed for selecting features to predict software module defects. Tree boosting algorithm namely LR, NB, KNN, NN, SVM, CART, RF, XGR have been used.

Here a systematic literature review [8] has been presented. The latest research has shown the area of Software Defect Prediction by discussing a critical review of papers published between 2016 and 2019. Initially, 1012 papers have been shortlisted from three online libraries (IEEE Xplore, ACM, and ScienceDirect); Out of these, 22 of these papers have been selected for detailed review in critical area.

III. CONTENT AND PROBLEM STATEMENT.

A lot of authors have worked ([1]-[8]) in the area of software defect prediction. Machine learning algorithms have also been proposed. However, no author has worked on the same data set and not evaluated several evaluation measures. That is the reason for this proposed work which has been written in this paper.

From the summary of literature review as mentioned in previous para, it has been found that a number of authors have tried to find out the machine learning models responsible for software defect protection. But they have not done which parameters are responsible for software defect prediction or not. If any parameter is responsible, how many percentage of contribution out of all parameters are responsible for software fault detection. In this paper, an effort is being made to find out which machine learning model is suitable for predicting software defect. Which parameters are responsible for eliminating software defect. If any software parameter is responsible for finding out predicting software fault., how much contribution of that parameter is responsible out of all software parameters exists. If that work is successful, preventive measures can be taken to eliminate software fault from software.

For this purpose Kaggle software defect data set[11] have been used. Machine learning models like Random Forest, Decision Tree, Logistic Regression, K-nearest neighbour, Gaussian Naïve Bayes, Support Vector Machine using linear function and Radial basis function, Gradient Boosting algorithm have been used. Accuracy, classification report and confidence matrix have been used as an evaluation parameter for selecting a particular machine learning model. Under confidence matrix, true positive numbers, false positive numbers, true negative numbers and false negative numbers have been used. Under classification report precision, recall, f1-score and support value have been used. After the selection of machine learning model based on accuracy, classification report and confidence matrix the particular machine learning model has to be used to find out the contribution of software parameters which are responsible for eliminating the fault of the software.

IV. METHODOLOGY.

A. Random Forest.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Random forests are an ensemble learning method for classification, regression. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Algorithm

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 and Step 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

B. Decision tree

Decision Trees are flowchart-like tree structures of all the possible solutions to a decision, based on certain conditions. It is called a decision tree as it starts from a root and then branches off to a number of decisions just like a tree. The tree starts from the root node where the most important attribute is placed. The branches represent a part of entire decision and each leaf node holds the outcome of the decision.

The best attribute or feature is selected using the Attribute Selection Measure (ASM). The attribute selected is the root node feature. Attribute selection measure is a technique used for the selecting best attribute for discrimination among tuples. It gives rank to each attribute and the best attribute is selected as splitting criterion. The most popular methods of selection are: (1) Entropy (2). Information Gain (3). Gain Ratio (4). Gini Index

1) *Entropy:* Entropy is the randomness in the information being processed. It measures the purity of the split. This algorithm computes the entropy with the following formula: $-(p \log_2(p)) -(q \log_2(q))$ p is the probability of success or the number of positive cases. q is the probability of failure or the number of negative cases.
2) *Information Gain:* Information gain is used to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned. Information gain tells us how important a given attribute of the feature vectors is. The ordering of attributes in the nodes of a decision tree can be decided with the help of this feature.
3) *Gain Ratio:* The gain ratio means the share of profit gained by a partner with some reconstitution of the firm. This gaining ratio is caused by the reconstitution which generally happens due to the exit or death of any existing partner.
4) *Gini Index:* Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with lower Gini index should be preferred.

The Formula for the calculation of the Gini Index is given below:-

Giniindex = $1 - \sum p_j^2$ where p_j is the probability of an object being classified to a particular class.

C. KNN(K Nearest Neighbours) algorithm

It is one of the simplest and widely used classification algorithms in which a new data point is classified based on similarity in the specific group of neighbouring data points.

Algorithm.

Step 1: Select the value of K neighbours
Step 2: Find the K nearest data point for our new data point based on euclidean distance
Step 3: Among these K data points count the data points in each category
Step 4: Assign the new data point to the category that has the most neighbours of the new data point.

D. Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm. It can be used in both classification and regression problems. Inherently, it is a discriminative classifier. Given a set of labelled data points, an SVM tries to separate the data points into different output classes. It does so by finding an optimal hyper plane that distinctly classifies the data points into an N-dimensional space (N - the number of features).

Support vectors are those two data points supporting the decision boundary (the data points which have the maximum margin from the hyper plane). Support Vector Machine (SVM) always makes an effort to those two data points from different classes that are the closest to each other. These support vectors are the keys to draw an optimal hyper plane by SVM. In SVM, the set of input and output data are treated as vectors. This is because when the data is a higher dimensional space (more than two dimensions), the classes cannot be represented as single data points, so they must be represented as vectors.

Non-linearly separable data can be separated by taking them into a higher dimension. It is necessary to map the data into just one dimension higher. Kernel Tricks are functions that apply on some complex mathematical operations on the lower-dimensional data points and convert them into higher dimensional space. Then finds out the process of separating the data points based on the labels and outputs. Common kernels are Linear Kernel, Polynomial Kernel, Radial Basis Function(RBF) of RBF Kernel, Sigmoid Kernel, Gaussian Kernel.

E. Logistic Regression.

Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is binary. Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables. The independent variables can be nominal, ordinal, or of interval type.

The name "logistic regression" is derived from the concept of the logistic function that it uses. The logistic function is also known as the sigmoid function. The value of this logistic function lies between zero and one.

F. Gaussian Nave Bayes Algorithm

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Naive Bayes is a basic but effective probabilistic classification model in machine learning that draws influence from Bayes Theorem.

Bayes theorem is a formula that offers a conditional probability of an event A for a given another event B has previously happened. Its mathematical formula is as follows: –

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Where A and B are two events

P(A|B) is the probability of event A provided event B has already happened.

P(B|A) is the probability of event B provided event A has already happened.

P(A) is the independent probability of A.

P(B) is the independent probability of B.

G. Gradient Boosting Algorithm

This algorithm starts by building a decision stump and then assigning equal weights to all the data points. Thereafter it increases the weights for all the points which are misclassified and lowers the weight for those that are easy to classify or are correctly classified. A new decision stump is made for these weighted data points. The idea behind this is to improve the predictions made by the first stump. The Decision Stump operator is used for generating a decision tree with only one single split. This operator can be very efficient when boosted with operators like the AdaBoost operator.

H. Confusion Matrix

The confusion matrix is a matrix which is used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. Since it shows the errors in the model performance in the form of a matrix, hence also known as an error matrix. Some features of Confusion matrix are given below:

For the 2 prediction classes of classifiers, the matrix is of 2*2 table, for 3 classes, it is 3*3 table, and so on.

The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.

Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.

The following table shows Confidence Matrix:

Total Predictions	Actual: No	Actual: Yes
Predicted: No	True Negative (TN)	False Positive (FP)
Predicted: Yes	False Negative (FN)	True Positive (TP)

The above table has the following cases:

1) *True Negative (TN):* Model has given prediction No, and the real or actual value was also No.
2) *True Positive (TP):* The model has predicted yes, and the actual value was also true. It is called Type-I error.
3) *False Negative (FP):* The model has predicted no, but the actual value was Yes, it is also called as Type-II error.
4) *False Positive (FP):* The model has predicted Yes, but the actual value was No.

It evaluates the performance of the classification models, predictions work on test data, and tells how good our classification model is. It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error. With the help of the confusion matrix, calculate the different parameters for the model, such as accuracy, precision, etc. can be calculated.

I. Classification Report

A classification report is a performance evaluation metric in machine learning. It is used to show the precision, recall, F1 Score, and support of the classification model. It provides a better understanding of the overall performance of our trained model.

1) *Precision:* Precision is defined as the ratio of true positives to the sum of true and false positives.
2) *Recall:* Recall is defined as the ratio of true positives to the sum of true positives and false negatives.
3) *F1 Score:* The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance of the model is.
4) *Support:* Support is the number of actual occurrences of the class in the dataset. It doesn't vary between models, it just diagnoses the performance evaluation process.

V. CONTRIBUTION.

A. Pre Processing of Dataset

Pre processing of data has to be done in order to remove the noisy data. This kind of data includes missing values, values which are out of range, null values, etc. The noisy data has to be smoothed out.

1) *Removal of noisy data:* Attribute values for a specific record may be blank or missing. These values are referred to as missing values. These data have to be removed for better accuracy of the model.
2) *Normalization:* The dataset may have different attributes which define the characteristics of the available records. Each of these attributes may be of different data types (like numerical, character, etc.). The range of values of each of these attributes may vary widely. Thus, the predictive model is likely to be biased toward the high or weighted values. In order to minimize this bias, all the values in the dataset are normalized (placed generally in the range of 0 to 1).
3) *Discretisation:* The dataset may have numerical values that are continuous in nature. These continuous values may not be able to predict the missing values. Thus, the predictive model makes use of the discretization technique to avoid such situations.

B. Classification

1) *Random Forest:* Input data set[11] has to be applied to random forest algorithm. It has used 10 estimators that means 10 decision trees have been constructed and finally the average of these tree values has to be taken. The value of accuracy has been found as 81.58 % for criterion as gini index as well as for entropy.
2) *Decision Tree:* Input data set [11] has to be applied to decision tree algorithm. Criterion as gini index has been used. The value of accuracy has been found as 81.57 %.
3) *KNN (K Nearest Neighbours) algorithm:* Input data set [11] has to be applied to KNN (K Nearest Neighbours algorithm. Number of neighbours has been used as 5. Distance function as 'minkowski' has been used. The value of accuracy has been found as 80.51 %.
4) *Support Vector Machine Algorithm:* Input data set [11] has to be applied to support vector machine algorithm. The value of accuracy has been found as 80.47 % based on kernel function as linear and as 31.58 % based on kernel function as radial basis function.
5) *Logistic Regression:* Input data set [11] has to be applied to logistic regression algorithm. The value of accuracy has been found as 81.21 %.
6) *Gaussian Naïve Bayes Algorithm:* Input data set [11] has to be applied to Gaussian Naïve Bayes algorithm. The value of accuracy has been found as 81.57 %.
7) *Gradient Boosting Classifier Algorithm:* Input data set [11] has to be applied to Gradient Boosting Classifier algorithm. The value of accuracy has been found as 78.94 %.

C. Training and Test data.

The number of training date has been used as 70% and that of test data as 30% for the above models for better performance of the models.

VI. RESULTS

The comparative study of all the models on the basis of accuracy, classification report and confidence matrix have been furnished in table 1, table 2, table 3 respectively.

Table 1
Machine Learning models versus accuracy

No	Name of Machine Learning Model	Accuracy(%)
1.	Random Forest Algorithm	81.58 %
2.	Decision Tree Algorithm	81.57 %
3.	KNN(K-Nearest Neighbour) Algorithm	80.51 %
4.	Support Vector Machine with Linear Kernel	80.47 %
5.	Support Vector Machine with Radial Basis FunctionKernel	31.58 %
6.	Logistic Regression Algorithm	81.21 %
7.	Gaussian Naïve Bayes Algorithm	81.57 %
8.	Gradient Boosting Classifier Algorithm	78.94 %

Table 2
Classification Report Item wise based on Machine Learning Models

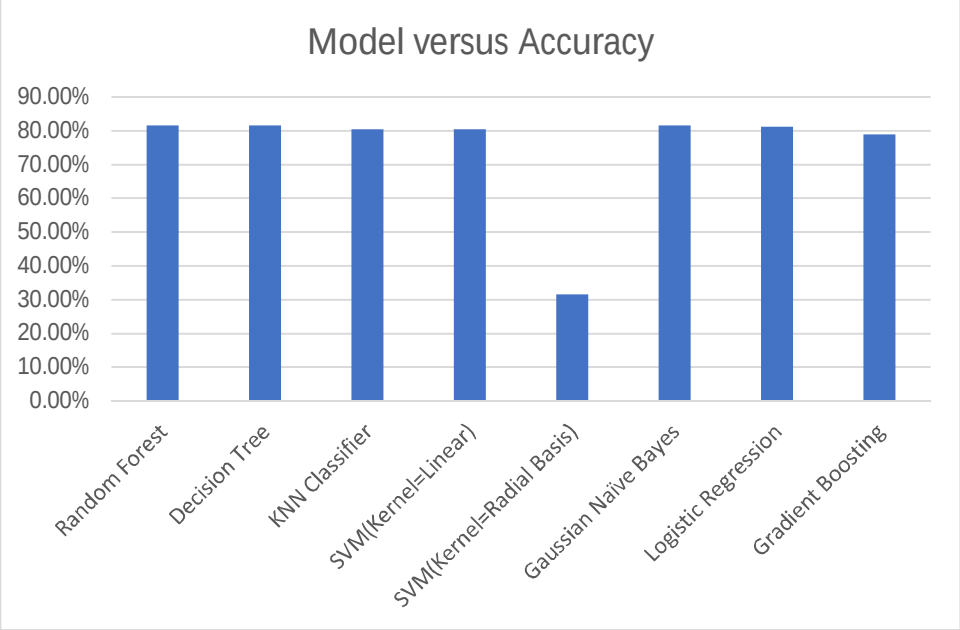
Model	Item	Precision	Recall	f1-score	Support
Random Forest	0	0.88	0.91	0.89	32
Random Forest	1	0.4	0.33	0.36	6
Decision Tree	0	0.88	0.91	0.89	32
Decision Tree	1	0.40	0.33	0.36	6
KNN Classifier	0	0.84	1.00	0.91	32
KNN Classifier	1	0.00	0.00	0.00	6
SVM(Kernel=Linear)	0	0.89	1.00	0.94	32
SVM(Kernel=Linear)	1	1.00	0.33	0.50	6
SVM(Kernel=Radial Basis)	0	0.93	0.81	0.87	32
SVM(Kernel=Radial Basis)	1	0.4	0.67	0.5	6
Gaussian Naïve Bayes	0	0.93	0.84	0.89	32
Gaussian Naïve Bayes	1	0.44	0.67	0.53	6
Logistic Regression	0	0.91	0.91	0.91	32
Logistic Regression	1	0.50	0.50	0.50	6
Gradient Boosting	0	0.93	0.81	0.87	32
Gradient Boosting	1	0.40	0.67	0.50	6

Table 3
Confusion Matrix based on Machine Learning Models

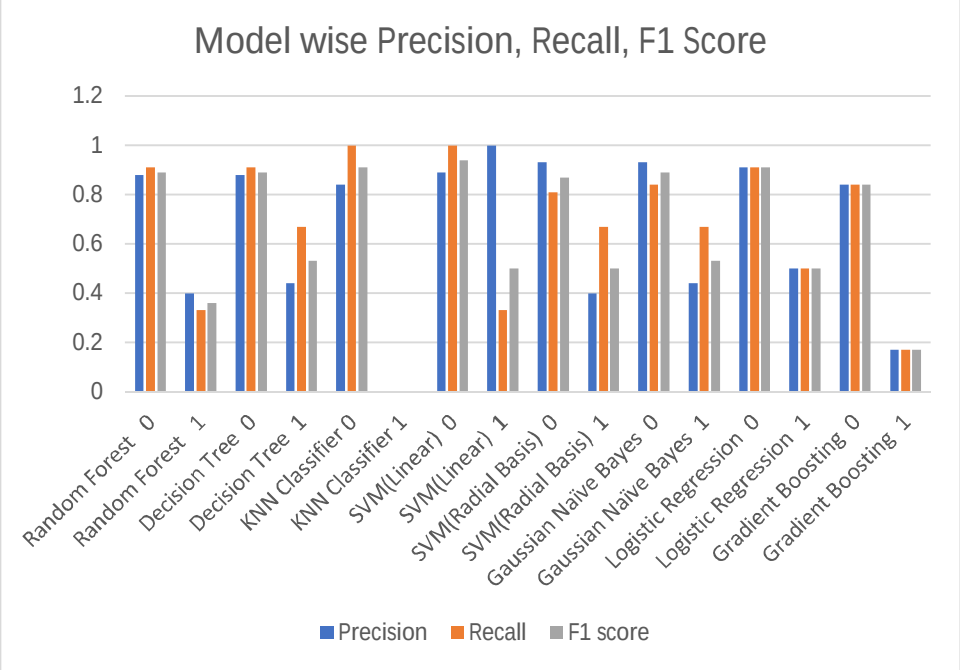
Model	True Positive	False Positive	False Negative	True Negative
Random Forest	32	0	0	6
Decision Tree	32	0	0	6
KNN Classifier	32	0	0	6
SVM(Kernel=Linear)	32	0	0	6
SVM(Kernel=Radial Basis)	32	0	0	6
Gaussian Naïve Bayes	32	0	0	6
Logistic Regression	32	0	0	6
Gradient Boosting	32	0	0	6

The preference of model has to be decided on the more value of accuracy, precision, recall, f1-score. From Table 1 it has been observed that the value of accuracy of Random Forest Algorithm is 81.58 % which is the maximum value among all algorithms. From Table 2 the value of precision, recall and f1-score is 1 in most cases for Random Forest as compared to other algorithms..

The Change of values of accuracy based on machine learning models have been furnished in graph named graph 1. The change of values of precision, recall, f1-acore based on machine learning models have been furnished in graph 2.



Graph 1



Graph 2

The contribution of various software parameters for improving the protection of software has been furnished in table 4. Here out of 21 software parameters in software data set [11], 10 parameters have appreciable contribution in providing fault free software where as other parameters have no effect/very feeble effect in software fault prediction.

Table 4
Contribution of software parameters in strengthening software fault prediction

No	Software Parameter	Contribution
1	rfc	28.07 %
2.	npm	8.86 %
3	cam	8.256
4.	moa	6.95 %
5.	loc	5.98%
6.	avg_cc	5.44 %
7	wmc	4.899 %
8	cbo	4.76 %
9	ce	4.69 %
10	lcom	4.469 %

VII. CONCLUSION

Random forest is a supervised learning algorithm. The forest that builds is an ensemble of decision trees, which is usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models which increases the accuracy of the overall result. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, in the domain of machine learning systems.

Random forest adds additional randomness to the model, during growing of the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally produces a better model.

Random forest utilizes the power of multiple decision trees. It depends on the feature importance given by a single decision tree. But the random forest chooses features randomly during the training process. Therefore, it does not depend highly on any specific set of features. This is a special characteristic of random forest over bagging trees. It can be mentioned Random Forest algorithm is suitable for situations when a large dataset is available. Since a random forest algorithm combines multiple decision trees, it becomes more difficult to explain. Random Forest has a higher training time than a single decision tree.

Random forest handles with the problem of overfitting by creating multiple trees, with each tree trained slightly differently so it overfits differently. Random forests is a classifier that combines a large number of decision trees. The decisions of each tree is then combined to make the final classification. This approach of random forest outperforms the “single generalist” approach of decision tree. Multiple overfitting classifiers are put together to reduce the overfitting.

By knowing the contribution of software dataset parameters involving in strengthening the software fault prediction, certain preventive measures can be taken in advance so that in future, software will not contain any fault.

REFERENCES

[1] Revoori Veeharika Reddy, Nagella Kedharnath, Mandi Akif Hussain, S. Vidya, “ Software Defect Estimation using Machine Learning Algorithms”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online), Volume-10 Issue-1, May 2021.

[2] N.Kalaivani , Dr.R.Beena, Overview of Software Defect Prediction using Machine Learning Algorithms, International Journal of Pure and Applied Mathematics, Volume 118 No. 20 2018, 3863-3873, ISSN: 1314-3395 (on-line version), url: http://www.ijpam.eu.

[3] Mitt Shah, Nandit Pujara, Software Defects Prediction Using Machine Learning. https://arxiv.org/ftp/arxiv/papers/2011/2011.00998.pdf.

[4] Wenjian Liu, Baoping Wang , and Wennan Wang, “Deep Learning Software Defect Prediction Methods for Cloud Environments Research”, Hindawi Scientific Programming, Volume 2021, Article ID 2323100, https://doi.org/10.1155/2021/2323100.

[5] Burcu Yal, Merve Ozedes, Software Defect Estimation Using Machine Learning Algorithms, (UBMK'19) 4rd International Conference on Computer Science and Engineering – 487, 978-1-7281-3964-7/19/$31.00@2019IEEE.

[6] Awni Hammouri, Mustafa Hammad, Mohammad Alnabhan, Fatima Alsarayrah, “ Software Bug Prediction using Machine Learning Approach”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 2, 2018.

[7] Geanderson Santos, Eduardo Figueiredo, Adriano Veloso, Markos Viggiato, Nivio Ziviani, “Predicting Software Defects with Explainable Machine Learning” SBQS’20, December 1–4, 2020, São Luís, Brazil, c 2020 Association for Computing Machinery. ACM ISBN 978-1-4503-8923-5/20/12. . . $15.00 https://doi.org/10.1145/3439961.3439979.

[8] Faseeha Matloob, Shabib Aftab, Munir Ahmad, Muhammad Adnan Khan, Areej Fatima, Muhammad Iqbal, Wesam Mohsen Alruwail and Nouh Sabri Elmitwally, "Software Defect Prediction Using Supervised Machine Learning Techniques: A Systematic Literature Review", Intelligent Automation & Soft Computing, DOI:10.32604/iasc.2021.017562.

[9] Dharmpal Singh, J. Paul Choudhury, Mallika De, "A Comparative Study of Meta Heuristic model to assess the type of Breast Cancer disease", IETE Journal of Research, June 2020, https://doi.org/10.1080/03772063-2020.1775139, Taylor & Francis Online. https://www.tandfonline.com/doi/abs/10.1080/03772063.2020.1775139?journalCode=tijr20#.X7qJOASky1M.gmail.

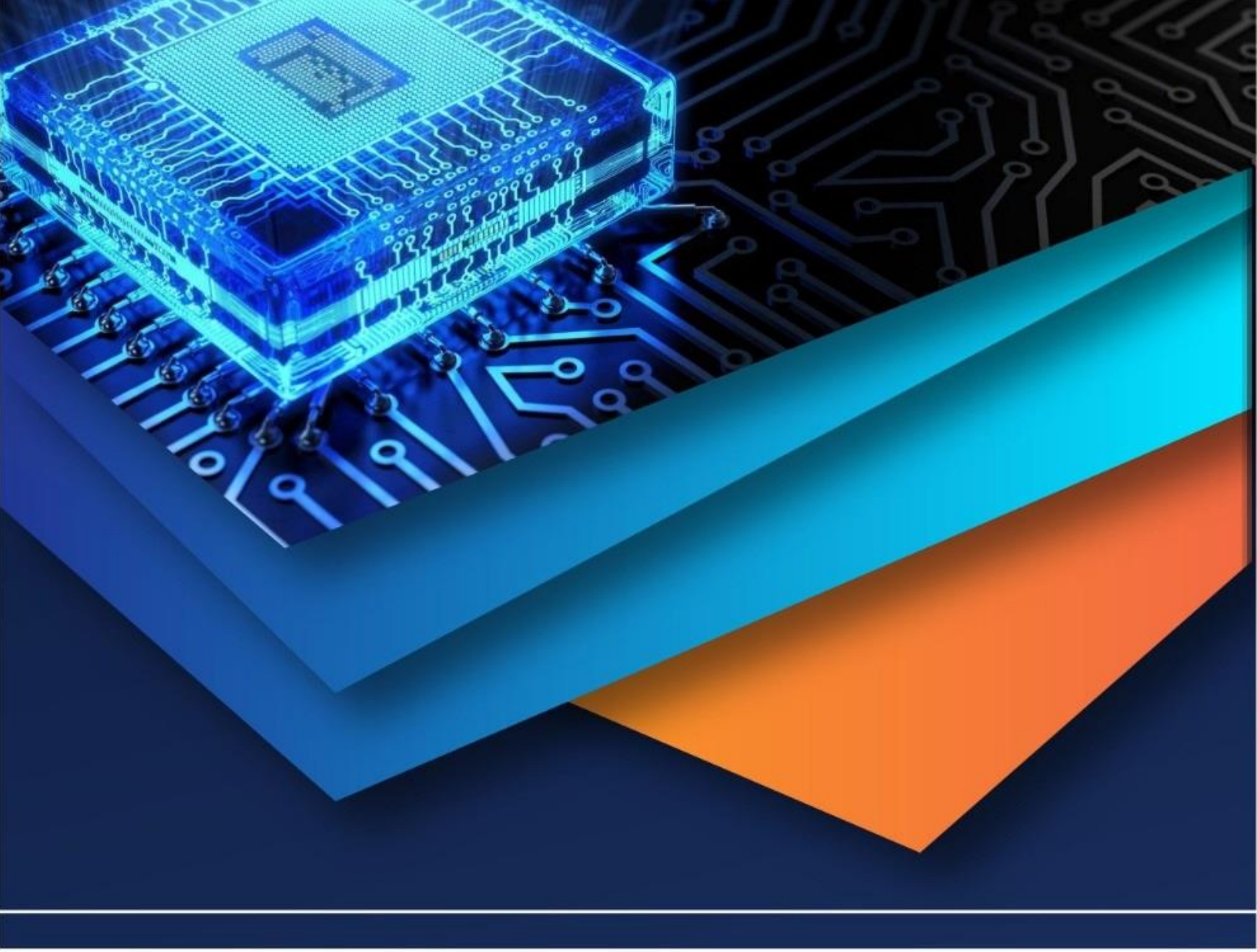
[10] Manisha Burman, J. Paul Choudhury, Susanta Biswas, "Automated Skin disease detection using multiclass PNN", International Journal of Innovations in Engineering &Technology(ISSN 2319-1058),http://dx.doi.org/10.21172/ijet 144.03, vol 14, issue 4, November 2019, pages 19-24.

[11] Software defect prediction data set(ant-1.3): https://www.kaggle.com/datasets/nazgolnikravesh/software-defect-prediction-dataset.

[12] Madhab Paul Choudhury, J. Paul Choudhury, "Intrusion Detection System using the method of Deep Learning and Clustering", Proceedings of 10 th International Conference on Frontiers of Intelligent Computing :Theory and Applications(FICTA 2022), NIT Mizoram, ISSN 2190-3018 ISSN 2190-3026 (electronic), Smart Innovation, Systems and Technologies ISBN 978-981-19-7512-7 ISBN 978-981-19-7513-4 (eBook) https://doi.org/10.1007/978-981-19-7513-4

[13] Madhab Paul Choudhury, J. Paul Choudhury, "A comparative study on the performance of Soft Computing models in the prediction of Orthopedic disease in the environment of Internet of Things". Proceedings of ICCM 2022 Department of Electronics and Communication Engineering and the Department of Electrical Engineering, North Eastern Regional Institute of Science and Technology (NERIST), Arunachal Pradesh, India in collaboration with Emlyon Business School France, July 2022 https://link.springer.com/chapter/10.1007/978-3-031-25194-8_20

[14] Madhab Paul Choudhury, J. Paul Choudhury, "Role of Soft Computing Models in Orthopedic Disease Prediction using Internet Of Things", Proceedings of 2022 AIMT GBU Spring International Conference on ICIRASMT,May 2022.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY